



SailPoint IdentityIQ Direct Connectors

Version 6.1 -1

Administration and Configuration Guide

© Copyright 2013 SailPoint Technologies, Inc., All Rights Reserved.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and reexport of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or reexport outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Entities List; a party prohibited from participation in export or reexport transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

Trademark Notices. Copyright © 2013 SailPoint Technologies, Inc. All rights reserved. SailPoint, the SailPoint logo, SailPoint IdentityIQ, and SailPoint Identity Analyzer are trademarks of SailPoint Technologies, Inc. and may not be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

Table of Contents

| | |
|---|-----------|
| Overview | 17 |
| Connector basics | 17 |
| Connector Licensing | 17 |
| Working of Connectors | 17 |
| Agent Connectors | 19 |
| Retryable mechanism | 19 |
| What's new in Direct Connectors | 20 |
| Application Types for Connectors | 20 |
| Viewing the available connectors | 23 |
| Connector selection | 23 |
| Section 1: Read/Write Direct Connectors | 25 |
| Chapter 1: SailPoint IdentityIQ Active Directory Connector | 27 |
| Overview | 27 |
| Supported features | 27 |
| Supported Managed System | 28 |
| Pre-requisites | 28 |
| Administrator permissions | 28 |
| Configuration parameters | 29 |
| Schema attributes | 30 |
| Account attributes | 30 |
| Group attributes | 35 |
| Provisioning Policy attributes | 36 |
| Install and register the IQService for Windows | 39 |
| Additional information | 40 |
| Unstructured Target Collector | 40 |
| Chapter 2: SailPoint IdentityIQ WebEx Connector | 43 |
| Overview | 43 |
| Supported features | 43 |
| Administrator permissions | 43 |
| Configuration parameters | 43 |
| Schema attributes | 44 |
| Account attributes | 44 |
| Group attributes | 46 |
| Provisioning Policy attributes | 46 |
| Chapter 3: SailPoint IdentityIQ Google Apps Connector | 49 |
| Overview | 49 |
| Supported features | 49 |
| Pre-requisites | 49 |
| Administrator permissions | 49 |
| Configuration parameters | 50 |
| Schema attributes | 50 |
| Account attributes | 50 |
| Group attributes | 50 |
| Provisioning Policy attributes | 51 |

| | |
|---|-----------|
| Chapter 4: SailPoint IdentityIQ LDAP Connector | 55 |
| Overview | 55 |
| Supported features | 55 |
| Supported Managed Systems | 56 |
| Pre-requisites | 57 |
| Administrator permissions | 58 |
| Configuration parameters | 58 |
| Schema attributes | 59 |
| Account attributes | 59 |
| Group attributes | 62 |
| Group Membership attribute | 63 |
| Group Entitlement attribute | 63 |
| Additional information | 64 |
| Support for NISNetGroups and POSIXGroups | 64 |
| Managing Revoke-Restore for SunOne | 65 |
| Using Novell eDirectory as a Pass-through Authentication Source | 66 |
| Chapter 5: SailPoint IdentityIQ Box.Net Connector | 67 |
| Overview | 67 |
| Supported features | 67 |
| Supported Managed Systems | 67 |
| Pre-requisites | 67 |
| Administrator permissions | 68 |
| Configuration parameter | 68 |
| Schema attributes | 68 |
| Account attributes | 68 |
| Group attributes | 68 |
| Provisioning Policy attributes | 69 |
| Chapter 6: SailPoint IdentityIQ Microsoft Office 365 Connector | 73 |
| Overview | 73 |
| Supported features | 73 |
| Prerequisites | 73 |
| Administrator permissions | 74 |
| Configuration parameters | 74 |
| Schema attributes | 75 |
| Account attributes | 75 |
| Group attributes | 76 |
| Provisioning Policy attributes | 77 |
| Install and register the IQService for Windows | 78 |
| Chapter 7: SailPoint IdentityIQ Microsoft Office 365 Exchange Online Connector | 79 |
| Overview | 79 |
| Supported features | 79 |
| Prerequisites | 79 |
| Administrator permissions | 80 |
| Configuration parameters | 80 |
| Schema attributes | 81 |
| Account attributes | 81 |
| Group attributes | 82 |
| Provisioning Policy attributes | 82 |
| Install and register the IQService for Windows | 84 |

Chapter 8: SailPoint IdentityIQ ServiceNow Connector85

| | |
|--------------------------------------|----|
| Overview | 85 |
| Supported features | 85 |
| Pre-requisites | 85 |
| Administrator permissions | 86 |
| Configuration parameters | 86 |
| Schema attributes | 87 |
| Account attributes | 88 |
| Group attributes | 89 |
| Provisioning Policy attributes | 90 |
| Additional information | 91 |
| Configuration settings | 91 |
| Troubleshooting | 92 |

Chapter 9: SailPoint IdentityIQ GoToMeeting Connector97

| | |
|--------------------------------------|----|
| Overview | 97 |
| Supported features | 97 |
| Pre-requisites | 97 |
| Administrator permissions | 97 |
| Configuration parameter | 97 |
| Schema attributes | 98 |
| Account attributes | 98 |
| Group attributes | 98 |
| Provisioning Policy attributes | 99 |

Chapter 10: SailPoint IdentityIQ Microsoft SharePoint Connector101

| | |
|--|-----|
| Overview | 101 |
| Supported features | 101 |
| Supported Managed system | 102 |
| Pre-requisites | 102 |
| Administrator permissions | 102 |
| Configuration parameters | 102 |
| Schema attributes | 103 |
| Account attributes | 103 |
| Group attributes | 103 |
| Provisioning Policy attributes | 104 |
| Install and register the IQService for Windows | 105 |
| Install and register the IQService | 105 |
| Additional information | 105 |
| Unstructured Target Collector | 105 |
| Troubleshooting | 107 |

Chapter 11: SailPoint IdentityIQ Amazon Web Services Identity and Access Management Connector109

| | |
|--------------------------------------|-----|
| Overview | 109 |
| Supported features | 110 |
| Pre-requisites | 110 |
| Administrator permissions | 111 |
| Schema attributes | 111 |
| Account schema | 111 |
| Group schema | 112 |
| Provisioning Policy attributes | 112 |
| Account | 112 |

| | |
|---|------------|
| Account-Group | 113 |
| Additional information | 113 |
| Amazon Web Services Identity and Access Management API's | 113 |
| Troubleshooting | 115 |
| Chapter 12: SailPoint IdentityIQ Microsoft SharePoint Online Connector | 117 |
| Overview | 117 |
| Supported features | 117 |
| Prerequisites | 117 |
| Administrator permissions | 118 |
| Configuration parameters | 118 |
| Schema attributes | 119 |
| Account attributes | 119 |
| Group attributes | 119 |
| Provisioning Policy attributes | 120 |
| Install and register the IQService for Windows | 121 |
| Additional information | 121 |
| Unstructured Target Collector | 121 |
| Chapter 13: SailPoint IdentityIQ NetSuite Connector | 123 |
| Overview | 123 |
| Supported features | 123 |
| Supported Managed Systems | 124 |
| Administrator permissions | 124 |
| Configuration parameters | 125 |
| Schema attributes | 125 |
| Account attributes | 125 |
| Group attributes | 126 |
| Schema extension and custom attributes | 126 |
| Provisioning Policy attributes | 127 |
| Additional information | 128 |
| NetSuite Application Program Interface (API) | 128 |
| Chapter 14: SailPoint IdentityIQ JDBC Connector | 129 |
| Overview | 129 |
| Supported features | 129 |
| Supported Managed Systems | 130 |
| Pre-requisites | 130 |
| Administrator permissions | 130 |
| Configuration parameters | 130 |
| Schema Attributes | 131 |
| JDBC Connector - Merging and Ordering | 131 |
| Chapter 15: SailPoint IdentityIQ PeopleSoft Connector | 133 |
| Overview | 133 |
| Supported features | 133 |
| Supported Managed Systems | 133 |
| Pre-requisites | 133 |
| Administrator permission | 134 |
| Configuration parameters | 134 |
| Schema attributes | 134 |
| Account attributes | 134 |
| Group attributes | 135 |

| | |
|--|------------|
| Additional information | 136 |
| Create the Component Interfaces | 136 |
| Create and Copy the required jar files | 136 |
| Configure Component Interface Security | 137 |
| Chapter 16: SailPoint IdentityIQ Siebel Connector | 139 |
| Overview | 139 |
| Supported features | 139 |
| Supported Managed Systems | 140 |
| Pre-requisites | 140 |
| Administrator permission | 140 |
| Configuration parameters | 140 |
| Schema attributes | 141 |
| Account attributes | 141 |
| Account Group attributes | 142 |
| Adding new custom attributes in schema | 143 |
| Provisioning policy attributes | 143 |
| Troubleshooting | 144 |
| Chapter 17: SailPoint IdentityIQ Lotus Domino Connector | 145 |
| Overview | 145 |
| Supported features | 145 |
| Supported Managed Systems | 146 |
| Pre-requisites | 146 |
| Administrator permissions | 147 |
| Configuration parameters | 147 |
| Schema attributes | 148 |
| Account attributes | 148 |
| Group attributes | 150 |
| Provisioning policy attributes | 150 |
| Create account attributes | 150 |
| Create group attributes | 152 |
| Update policies | 152 |
| Install and register the IQService | 154 |
| Additional information | 154 |
| ID Vault functionalities | 155 |
| Password management | 155 |
| Troubleshooting | 155 |
| Chapter 18: SailPoint IdentityIQ Microsoft SQL Server | 157 |
| Overview | 157 |
| Supported features | 157 |
| Supported Managed Systems | 158 |
| Pre-requisites | 158 |
| Administrator permissions | 158 |
| Configuration parameters | 158 |
| Schema attributes | 159 |
| Account attributes | 159 |
| Group attributes | 160 |
| Provisioning Policy attributes | 160 |
| Additional information | 160 |
| Delete login | 161 |
| Direct permission | 161 |

| | |
|---|------------|
| Identity and Entitlement representation | 161 |
| Chapter 19: SailPoint IdentityIQ Oracle Connector | 163 |
| Overview | 163 |
| Supported features | 163 |
| Supported Managed Systems | 163 |
| Pre-requisites | 164 |
| Administrator permissions | 164 |
| Configuration parameters | 165 |
| Schema attributes | 165 |
| Account attributes | 165 |
| Group attributes | 166 |
| Provisioning policy attributes | 166 |
| Troubleshooting | 167 |
| Chapter 20: SailPoint IdentityIQ Sybase Connector | 169 |
| Overview | 169 |
| Supported features | 169 |
| Supported Managed Systems | 170 |
| Pre-requisites | 170 |
| Administrator permissions | 170 |
| Configuration parameters | 170 |
| Schema attributes | 171 |
| Account attributes | 171 |
| Group attributes | 172 |
| Provisioning policy attributes | 172 |
| Additional information | 172 |
| Delete login | 173 |
| Direct Permissions | 173 |
| Identity and Entitlement representation | 173 |
| Troubleshooting | 174 |
| Chapter 21: SailPoint IdentityIQ Windows Local Connector | 175 |
| Overview | 175 |
| Supported features | 175 |
| Supported Managed Systems | 176 |
| Pre-requisites | 176 |
| Administrator permissions | 176 |
| Configuration parameters | 176 |
| Schema attributes | 177 |
| Account attributes | 177 |
| Group attributes | 178 |
| Provisioning Policy attributes | 178 |
| Install and register IQService | 179 |
| Additional information | 179 |
| Unstructured Target Collector | 180 |
| Troubleshooting | 181 |
| Chapter 22: SailPoint IdentityIQ AIX Connector | 183 |
| Overview | 183 |
| Supported features | 183 |
| Supported Managed Systems | 184 |
| Pre-requisites | 184 |

| | |
|---|------------|
| Administrator permissions | 184 |
| Configuration parameters | 184 |
| Additional configuration parameters for SSH configuration | 185 |
| Public key authentication configuration | 185 |
| Schema attributes | 186 |
| Account attributes | 186 |
| Group attributes | 192 |
| Provisioning policy attributes | 192 |
| Account attributes | 192 |
| Group attributes | 193 |
| Additional information | 193 |
| Unstructured Target Collector | 194 |
| Troubleshooting | 194 |
| Chapter 23: SailPoint IdentityIQ Linux Connector | 197 |
| Overview | 197 |
| Supported features | 197 |
| Supported Managed Systems | 198 |
| Pre-requisites | 198 |
| Administrator permissions | 198 |
| Configuration parameters | 198 |
| Additional configuration parameters for SSH configuration | 199 |
| Public key authentication configuration | 199 |
| Schema attributes | 200 |
| Account attributes | 200 |
| Group attributes | 201 |
| Provisioning policy attributes | 201 |
| Account attributes | 201 |
| Group attributes | 202 |
| Additional information | 202 |
| Unstructured Target Collector | 202 |
| Troubleshooting | 203 |
| Chapter 24: SailPoint IdentityIQ Solaris Connector | 207 |
| Overview | 207 |
| Supported features | 207 |
| Supported Managed Systems | 208 |
| Pre-requisites | 208 |
| Administrator permissions | 208 |
| Configuration parameters | 209 |
| Additional configuration parameters for SSH configuration | 209 |
| Public key authentication configuration | 209 |
| Schema attributes | 210 |
| Account attributes | 210 |
| Group attributes | 212 |
| Provisioning policy attributes | 212 |
| Account attributes | 212 |
| Group attributes | 213 |
| Additional information | 213 |
| Unstructured Target Collector | 214 |
| Troubleshooting | 214 |

| | |
|--|------------|
| Chapter 25: SailPoint IdentityIQ BMC Remedy IT Service Management Suite Connector | 219 |
| Overview | 219 |
| Supported features | 219 |
| Supported Managed Systems | 220 |
| Pre-requisites | 220 |
| Administrator permission | 220 |
| Configuration parameters | 220 |
| Schema attributes | 220 |
| Account attributes | 220 |
| Group attributes | 221 |
| Provisioning policy attributes | 222 |
| Create account attributes | 222 |
| Create group attributes | 223 |
| Update policies | 223 |
| Additional information | 224 |
| Enable/Disable Account | 224 |
| Add Entitlement operation for ITSM | 224 |
| Troubleshooting | 224 |
| Chapter 26: SailPoint IdentityIQ Jive Connector | 225 |
| Overview | 225 |
| Supported features | 225 |
| Pre-requisites | 225 |
| Administrator permission | 226 |
| Configuration parameters | 226 |
| Schema attributes | 226 |
| Account attributes | 226 |
| Group attributes | 227 |
| Provisioning Policy attributes | 228 |
| Create account attributes | 228 |
| Create group attributes | 228 |
| Chapter 27: SailPoint IdentityIQ Oracle E-Business Suite Connector | 229 |
| Overview | 229 |
| Supported features | 229 |
| Supported Managed Systems | 230 |
| Pre-requisites | 230 |
| Administrator permissions | 230 |
| Configuration parameters | 231 |
| Schema attributes | 231 |
| Account attributes | 231 |
| Group attributes | 232 |
| Provisioning Policy attributes | 232 |
| Create account attributes | 233 |
| Delete account attributes | 233 |
| Create group attributes | 233 |
| Deleting Group (Responsibility) | 234 |
| Deleting entitlement | 234 |
| Troubleshooting | 234 |

| | |
|--|------------|
| Chapter 28: SailPoint IdentityIQ Rally Connector | 235 |
| Overview | 235 |
| Supported features | 235 |
| Pre-requisites | 235 |
| Administrator permission | 235 |
| Configuration parameters | 235 |
| Schema attributes | 236 |
| Provisioning Policy attributes | 236 |
| Chapter 29: SailPoint IdentityIQ BMC Remedy Connector | 239 |
| Overview | 239 |
| Supported features | 239 |
| Supported Managed Systems | 239 |
| Pre-requisites | 240 |
| Administrator permission | 240 |
| Configuration parameters | 240 |
| Schema attributes | 240 |
| Account attributes | 240 |
| Group attributes | 241 |
| Provisioning policy attributes | 241 |
| Create account attributes | 242 |
| Create group attributes | 242 |
| Update policies | 242 |
| Additional information | 242 |
| Enable/Disable Account | 242 |
| Troubleshooting | 243 |
| Chapter 30: SailPoint IdentityIQ RSA Ace Server Connector | 245 |
| Overview | 245 |
| Supported features | 245 |
| Supported Managed Systems | 245 |
| Pre-requisites | 246 |
| Administrator permissions | 247 |
| Configuration parameters | 247 |
| Schema attributes | 248 |
| Account attributes | 248 |
| Group attributes | 248 |
| Provisioning Policy attributes | 249 |
| Chapter 31: SailPoint IdentityIQ Salesforce/Remedyforce Connector | 251 |
| Overview | 251 |
| Supported features | 251 |
| Administrator permissions | 252 |
| Configuration parameters | 252 |
| Schema attributes | 253 |
| Account attributes | 253 |
| Profile attributes | 255 |
| Provisioning Policy attributes | 256 |
| Chapter 32: SailPoint IdentityIQ SAP Connector | 257 |
| Overview | 257 |
| Supported features | 257 |
| Supported Managed Systems | 258 |

| | |
|---|------------|
| Pre-requisites | 258 |
| Administrator permissions | 258 |
| Configuration parameters | 258 |
| Schema attributes | 259 |
| Account attributes | 259 |
| Group attributes | 262 |
| Schema extension and custom attributes | 263 |
| Provisioning Policy attributes | 263 |
| Create account attributes | 263 |
| Additional information | 263 |
| Entitlement validity period | 264 |
| CUA support | 264 |
| Troubleshooting | 264 |
| Chapter 33: SailPoint IdentityIQ SAP Enterprise Portal Connector | 265 |
| Overview | 265 |
| Supported features | 265 |
| Supported Managed Systems | 266 |
| Prerequisites | 266 |
| Administrator permission | 266 |
| Configuration parameters | 266 |
| Schema attributes | 267 |
| Account attributes | 267 |
| Group attributes | 268 |
| Provisioning Policy attributes | 268 |
| Create account attributes | 268 |
| Create Group attributes | 269 |
| Chapter 34: SailPoint IdentityIQ Tivoli Access Manager Connector | 271 |
| Overview | 271 |
| Supported features | 271 |
| Supported Managed System | 271 |
| Pre-requisites | 271 |
| Configuration parameters | 272 |
| Schema attributes | 273 |
| Account attributes | 273 |
| Group attributes | 273 |
| Provisioning Policy attributes | 274 |
| Create account attributes | 274 |
| Create group attributes | 274 |
| Additional information | 275 |
| Unstructured Target Collector | 275 |
| Troubleshooting | 275 |
| Chapter 35: SailPoint IdentityIQ Tenrox Connector | 277 |
| Overview | 277 |
| Supported features | 277 |
| Supported Managed Systems | 277 |
| Pre-requisites | 278 |
| Administrator permission | 278 |
| Configuration parameters | 278 |
| Schema attributes | 278 |
| Account attributes | 278 |

| | |
|---|------------|
| Provisioning Policy attributes | 279 |
| Troubleshooting | 280 |
| Section 2: Read Only Direct Connectors | 281 |
| Chapter 36: SailPoint IdentityIQ Yammer Connector | 283 |
| Overview | 283 |
| Pre-requisites | 283 |
| Configuration parameter | 283 |
| Schema attributes | 283 |
| Account attributes | 283 |
| Group attributes | 284 |
| Chapter 37: SailPoint IdentityIQ ALES Connector | 287 |
| Overview | 287 |
| Configuration parameters | 287 |
| Schema attributes | 288 |
| Account attributes | 288 |
| Group attributes | 288 |
| Chapter 38: SailPoint IdentityIQ Logical Connector | 289 |
| Overview | 289 |
| Configuration parameters | 289 |
| Schema attributes | 289 |
| Additional information | 290 |
| Logical Connector - Tiers Tab | 290 |
| Defining Logical Connectors | 292 |
| Logical Application Filtering | 292 |
| Chapter 39: SailPoint IdentityIQ Delimited Connector | 293 |
| Overview | 293 |
| Configuration parameters | 293 |
| Schema attributes | 295 |
| Chapter 40: SailPoint IdentityIQ LDIF Connector | 297 |
| Overview | 297 |
| Configuration parameters | 297 |
| Schema Attributes | 298 |
| Account attributes | 298 |
| Group attributes | 302 |
| Chapter 41: SailPoint IdentityIQ IBM Tivoli Identity Manager Connector | 303 |
| Overview | 303 |
| Configuration parameters | 303 |
| Schema attributes | 304 |
| Account attributes | 304 |
| Group attributes | 307 |
| Chapter 42: SailPoint IdentityIQ SAP HR/HCM Connector | 309 |
| Overview | 309 |
| Configuration parameters | 309 |
| Schema Attributes | 309 |
| Account attributes | 310 |

| | |
|---|------------|
| Chapter 43: SailPoint IdentityIQ Sun IDM Connector | 315 |
| Overview | 315 |
| Configuration parameters | 315 |
| Chapter 44: SailPoint IdentityIQ Top Secret Connector | 317 |
| Overview | 317 |
| Configuration parameters | 317 |
| Schema Attributes | 318 |
| Chapter 45: SailPoint IdentityIQ UNIX Connector | 331 |
| Overview | 331 |
| Configuration parameters | 331 |
| Schema attributes | 331 |
| Account attributes | 332 |
| Group attributes | 332 |
| Chapter 46: SailPoint IdentityIQ Mainframe Connector | 333 |
| Overview | 333 |
| Configuration parameters | 333 |
| Schema attributes | 334 |
| Account attributes | 334 |
| Chapter 47: SailPoint IdentityIQ Novell Identity Manager Connector | 335 |
| Overview | 335 |
| Configuration parameters | 335 |
| Schema attributes | 336 |
| Account attributes | 336 |
| Group attributes | 339 |
| Chapter 48: SailPoint IdentityIQ RACF Connector | 341 |
| Overview | 341 |
| Configuration parameters | 341 |
| Schema Attributes | 342 |
| Account attributes | 342 |
| Group attributes | 346 |
| Chapter 49: SailPoint IdentityIQ Rule Based Logical Connector | 349 |
| Overview | 349 |
| Configuration parameters | 349 |
| Section 3: Appendix Section | 351 |
| Appendix A: Password Interceptor 353 | |
| Password Interceptor on IdentityIQ Server | 353 |
| Password Intercept Web Service | 353 |
| Password Intercept Workflow | 353 |
| Installation | 354 |
| Uninstallation | 358 |
| Password Interceptor for LDAP | 358 |
| Installation | 358 |
| Uninstallation | 362 |
| Password Interceptor for UNIX | 363 |
| (Only for AIX) Using aix-pwi-set.sh script | 363 |

| | |
|---|-----|
| Installation | 364 |
| Verifying the Password Interceptor installation | 367 |
| Uninstallation | 370 |
| Managing Password Interceptor messages | 370 |
| Manually Start/Stop Password Interceptor Client | 371 |
| Troubleshooting | 372 |
| Appendix B: IQService Before/After Script 373 | |
| Overview | 373 |
| Writing a script | 374 |
| Scripts with Object Oriented support | 374 |
| Scripts without Object Oriented support | 376 |
| Creating a Rule | 376 |
| Configuring the Rules in Application | 377 |
| Appendix C: Delta Aggregation 379 | |
| Overview | 379 |
| Delta aggregation for Microsoft Active Directory, ADAM, SunOne and Tivoli | 379 |
| Configuring server for Delta Aggregation | 380 |
| Testing Delta Aggregation | 380 |

Overview

The following topics are discussed:

| | |
|--|----|
| Connector basics | 17 |
| Retryable mechanism | 19 |
| What's new in Direct Connectors | 20 |
| Application Types for Connectors | 20 |
| Viewing the available connectors | 23 |

This document describes the different types of connectors available for integrating between external applications and IdentityIQ, as well as the process required to install and configure those connectors.

- Note:** To enable group provisioning for existing application in **IdentityIQ** after upgrading **IdentityIQ** to version 6.1, perform the following:
- add **GROUP_PROVISIONING** to the featureString from **IdentityIQ** debug page
 - Define **CreateGroup** and **EditGroup** provisioning policies. For more information on the provisioning policies defined for the connector from **connectorRegistry.xml**, see the "Provisioning Policy" section of the respective connectors for the required attributes.

Connector basics

IdentityIQ makes use of several different types of connectors. Connectors are commonly grouped by the ways in which they can communicate with IdentityIQ. There are:

- read-only connectors that can only communicate data into IdentityIQ from an external application (Governance)
- read-write connectors that can read data from external applications and write data out to them (Gateway and Direct)

Connector Licensing

All IdentityIQ customers are automatically licensed to use any of the Governance connectors. However, the Direct, Gateway, and Agent connectors are separately licensed with the product's Provisioning Engine. Many of the Direct connectors began as Governance connectors and were modified to add the Provisioning capabilities. Customers who have not licensed the Provisioning Engine may use connectors for reading data (aggregation) but are not permitted to implement the provisioning features of connectors without purchasing a Provisioning Engine license.

Working of Connectors

This section describes how the connectors work.

Governance Connector

Governance connectors are very simple in design; they make a direct read-only connection to the external application through the connection parameters specified on the Application Definition.

The currently available Governance Connectors are listed below:

- LDIF
- SAP HR/HCM

Connector basics

- UNIX
- VMS
- Mainframe
- TopSecret
- Delimited File
- Logical
- RuleBasedFileParser
- RuleBasedLogical
- Yammer

Direct Connectors

Direct connectors are read-write connectors that allow IdentityIQ and the external application to send data directly between them in both directions. When read and write capabilities are needed for applications that have these connectors available, they are the most efficient and best choice to implement.

The current set of direct connectors are listed below:

- ADAM - Direct
- JDBC
- Novell Edirectory - Direct
- OID - Direct
- OpenLDAP - Direct
- SunOne - Direct
- Tivoli - Direct
- Google Apps
- Webex
- Salesforce
- Active Directory
- GotoMeeting
- Box.NET
- NetSuite
- AWS
- Office 365
- SharePoint Online
- Exchange Online
- SharePoint Inpremises
- IBM Lotus Domino
- BMC Remedy IT Service Management
- BMC Remedy
- Oracle E-Business Suite
- RSA Ace Server
- SAP
- SAP Enterprise Portal
- Tenrox

- Rally
- Tivoli Access Manager
- ServiceNow
- Microsoft SQL Server
- Oracle
- AIX
- Linux
- Solaris
- Sybase
- PeopleSoft
- RemedyForce

Gateway Connectors

Gateway connectors which connect to the external application via Connector Manager have been re-written as Direct Connectors. Please see the complete list in “Direct Connectors” on page 18. Also refer to benefits of Direct Connectors under “What’s new in Direct Connectors” on page 20.

Agent Connectors

The targeted systems for Agent connectors are centralized mainframe security systems; Agents are the simplest and most secure way to connect to those systems. Like the Gateway connectors, Agents communicate with IdentityIQ through the Connector Gateway. In Agent connectors, the functionality of the Connector Manager is contained within the Agent, so the Connector Manager is not required.

Following are the Agent Connectors for IdentityIQ:

- ACF2
- AS400
- RACF Full
- TopSecret Full
- DB2-UDB

Target permissions support (RACF, ACF2, and Top Secret)

The Target permissions feature is supported for Mainframe based connectors that include RACF, ACF2 and Top Secret.

For more information on target permissions, see *SailPoint IdentityIQ Integration Guide version 6.1*.

Retryable mechanism

For availing the advantage of some of the logic around retryable situations, add the retryable error messages list to the attributes map on an application. The **retryableErrors** entry is a list of strings through which the connector searches when it receives a message from the managed application. If one of the strings in the entry exists in the error, the connector attempts to retry the connection. When the configured error string is not a part of the error message returned from the connector, then IdentityIQ will not attempt a retry.

Here is an example of this entry:

What's new in Direct Connectors

```
<entry key="retryableErrors">
  <value>
    <List>
      <String>Server is not operational</String>
    </List>
  </value>
</entry>
```

Precaution: Avoid using error messages which contain a date/time, sequence id, SM packets/messages, and so on, as these are very specific. Error codes or error message substrings would be good candidates for inclusion.

What's new in Direct Connectors

- **Simple to configure and use:** Direct connectors are simple to configure, few configuration details required to use the connector and no extra steps to deploy agents on the end managed systems.
- **Less moving parts:** Direct connectors do not require Connector Gateway (CG), Connector Manager (CM), Provisioning Modules (PM) to be deployed to get the setup done. Installing and (Re)configuring each component is not required. Data caching and sequencing on transactions not required.
- **Increased performance:** The performance of direct connectors is improved compared to the old FULL or Gateway based connectors. It is recommended to move to direct connector to get maximum benefit of per transactions.
- **No single point of failure:** Earlier if one component failed in the connector model, then it required re-cycling of the Connector and Connector Gateway. Such issues do not exist in direct connector architecture.
- **Less hardware:** New direct connectors do not require any agent installation on end managed system or other computer. The overall hardware requirement for IdentityIQ and connectors setup is reduced due to this new architecture.

Application Types for Connectors

The following table lists the application type of Connectors in IdentityIQ:

Table 1—Application Types for Connectors

| Connector | Application Type | Details |
|------------------|---------------------------|---|
| Active Directory | Active Directory - Direct | Chapter 1: SailPoint IdentityIQ Active Directory Connector 27 |
| Webex | Webex | Chapter 2: SailPoint IdentityIQ WebEx Connector 43 |
| Google Apps | GoogleApps | Chapter 3: SailPoint IdentityIQ Google Apps Connector 49 |
| LDAP | LDAP | Chapter 4: SailPoint IdentityIQ LDAP Connector 55 |
| Box.Net | Box.Net | Chapter 5: SailPoint IdentityIQ Box.Net Connector 67 |

Table 1—Application Types for Connectors

| Connector | Application Type | Details |
|--|-------------------------------|---|
| Microsoft Office 365 | Microsoft Office365 | Chapter 6: SailPoint IdentityIQ Microsoft Office 365 Connector 73 |
| Microsoft Office 365 Exchange Online | Microsoft Exchange Online | Chapter 7: SailPoint IdentityIQ Microsoft Office 365 Exchange Online Connector 79 |
| ServiceNow | ServiceNow | Chapter 8: SailPoint IdentityIQ ServiceNow Connector 85 |
| GoToMeeting | GoToMeeting | Chapter 9: SailPoint IdentityIQ GoToMeeting Connector 97 |
| Microsoft SharePoint Connector | Microsoft SharePoint | Chapter 10: SailPoint IdentityIQ Microsoft SharePoint Connector 101 |
| Amazon Web Services Identity and Access Management | AWS IAM | Chapter 11: SailPoint IdentityIQ Amazon Web Services Identity and Access Management Connector 109 |
| Microsoft SharePoint Online | Microsoft SharePoint Online | Chapter 12: SailPoint IdentityIQ Microsoft SharePoint Online Connector 117 |
| NetSuite | NetSuite | Chapter 13: SailPoint IdentityIQ NetSuite Connector 123 |
| JDBC | JDBC | Chapter 14: SailPoint IdentityIQ JDBC Connector 129 |
| PeopleSoft | PeopleSoft - Direct | Chapter 15: SailPoint IdentityIQ PeopleSoft Connector 133 |
| Siebel | Siebel | Chapter 16: SailPoint IdentityIQ Siebel Connector 139 |
| Lotus Domino | IBM Lotus Domino - Direct | Chapter 17: SailPoint IdentityIQ Lotus Domino Connector 145 |
| Microsoft SQL Server | Microsoft SQL Server - Direct | Chapter 18: SailPoint IdentityIQ Microsoft SQL Server 157 |
| Oracle | Oracle Database - Direct | Chapter 19: SailPoint IdentityIQ Oracle Connector 163 |
| Sybase | Sybase - Direct | Chapter 20: SailPoint IdentityIQ Sybase Connector 169 |
| Windows Local | Windows Local - Direct | Chapter 21: SailPoint IdentityIQ Windows Local Connector 175 |
| AIX | AIX - Direct | Chapter 22: SailPoint IdentityIQ AIX Connector 183 |
| Linux | Linux - Direct | Chapter 23: SailPoint IdentityIQ Linux Connector 197 |

Table 1—Application Types for Connectors

| Connector | Application Type | Details |
|-----------------------------|-----------------------------------|---|
| Solaris | Solaris - Direct | Chapter 24: SailPoint IdentityIQ Solaris Connector 207 |
| Remedy ITSM | BMC ITSM - Direct | Chapter 25: SailPoint IdentityIQ BMC Remedy IT Service Management Suite Connector 219 |
| Jive | JIVE | Chapter 26: SailPoint IdentityIQ Jive Connector 225 |
| Oracle EBS | Oracle Applications - Direct | Chapter 27: SailPoint IdentityIQ Oracle E-Business Suite Connector 229 |
| Rally | Rally | Chapter 28: SailPoint IdentityIQ Rally Connector 235 |
| Remedy | BMC Remedy - Direct | Chapter 29: SailPoint IdentityIQ BMC Remedy Connector 239 |
| RSA ACE Server | RSA Ace Server - Direct | Chapter 30: SailPoint IdentityIQ RSA Ace Server Connector 245 |
| Salesforce/Remedyforce | RemedyForce | Chapter 31: SailPoint IdentityIQ Salesforce/Remedyforce Connector 251 |
| SAP | SAP - Direct | Chapter 32: SailPoint IdentityIQ SAP Connector 257 |
| SAP EP | SAP Portal - UMWebService | Chapter 33: SailPoint IdentityIQ SAP Enterprise Portal Connector 265 |
| Tivoli Access Manager | IBM Tivoli Access Manager | Chapter 34: SailPoint IdentityIQ Tivoli Access Manager Connector 271 |
| Tenrox | Tenrox | Chapter 35: SailPoint IdentityIQ Tenrox Connector 277 |
| Yammer | Yammer | Chapter 36: SailPoint IdentityIQ Yammer Connector 283 |
| ALES | BEA Aqualogic Enterprise Security | Chapter 37: SailPoint IdentityIQ ALES Connector 287 |
| Logical | Logical | Chapter 38: SailPoint IdentityIQ Logical Connector 289 |
| Delimited | DelimitedFile | Chapter 39: SailPoint IdentityIQ Delimited Connector 293 |
| LDIF | LDIF | Chapter 40: SailPoint IdentityIQ LDIF Connector 297 |
| IBM Tivoli Identity Manager | IBM Tivoli Identity Manager | Chapter 41: SailPoint IdentityIQ IBM Tivoli Identity Manager Connector 303 |

Table 1—Application Types for Connectors

| Connector | Application Type | Details |
|-------------------------|-------------------------|--|
| SAP HR/HCM | SAP HR/HCM | Chapter 42: SailPoint IdentityIQ SAP HR/HCM Connector 309 |
| Sun IDM | Sun IDM | Chapter 43: SailPoint IdentityIQ Sun IDM Connector 315 |
| Top Secret | TopSecret | Chapter 44: SailPoint IdentityIQ Top Secret Connector 317 |
| UNIX | Unix | Chapter 45: SailPoint IdentityIQ UNIX Connector 331 |
| Mainframe | Mainframe | Chapter 46: SailPoint IdentityIQ Mainframe Connector 333 |
| Novell Identity Manager | Novell Identity Manager | Chapter 47: SailPoint IdentityIQ Novell Identity Manager Connector 335 |
| RACF | RACF | Chapter 48: SailPoint IdentityIQ RACF Connector 341 |
| Rule Based Logical | RuleBasedFileParser | Chapter 49: SailPoint IdentityIQ Rule Based Logical Connector 349 |

Viewing the available connectors

Connectors may be added, removed, or modified in any IdentityIQ release, including patch releases. Existing defined applications will continue to use the connector specified during their initial creation, and changes to the connector will not affect existing applications unless those changes are manually applied to the application definition. However, the ConnectorRegistry entry for the connectors does change with new releases. The list of available connectors, with their current set of available features, can be retrieved from the Connector Registry within the IdentityIQ Debug Pages.

Select **Configuration** in the Objects list and click **List**, then select **ConnectorRegistry** to view the XML for all the connectors.

The **featuresString** value on each connector indicates the functionality that connector is capable of providing; when **PROVISIONING** is specified in the **featuresString**, the connector is a write-capable connector. The attribute "<entry key="MscsType" value="[MSCS-Type-Name]"/>" tells the name to specify for that connector's MSCS Type value (also listed in the previous section here).

The out-of-the-box connector specifications can also be found in the **ConnectorRegistry.xml** file in the [IdentityIQ Installation Directory]/WEB-INF/bin directory.

Connector selection

Often there is more than one connector that can communicate with a single external application, which may raise questions as to which one is the best choice. When multiple connectors exist for a single application, they are always of different types. The “best” choice is dictated by the needs (and license limitations) of the organization.

Viewing the available connectors

More information on connector selection is provided in the introductory section for each application within this document.

Section 1: Read/Write Direct Connectors

This section contains the information on the following:

- SailPoint IdentityIQ Active Directory Connector on page 27
- SailPoint IdentityIQ WebEx Connector on page 43
- SailPoint IdentityIQ Google Apps Connector on page 49
- SailPoint IdentityIQ LDAP Connector on page 55
- SailPoint IdentityIQ Box.Net Connector on page 67
- SailPoint IdentityIQ Microsoft Office 365 Connector on page 73
- SailPoint IdentityIQ Microsoft Office 365 Exchange Online Connector on page 79
- SailPoint IdentityIQ ServiceNow Connector on page 85
- SailPoint IdentityIQ GoToMeeting Connector on page 97
- SailPoint IdentityIQ Microsoft SharePoint Connector on page 101
- SailPoint IdentityIQ Microsoft SharePoint Online Connector on page 117
- SailPoint IdentityIQ Amazon Web Services Identity and Access Management Connector on page 109
- SailPoint IdentityIQ NetSuite Connector on page 123
- SailPoint IdentityIQ JDBC Connector on page 129
- SailPoint IdentityIQ PeopleSoft Connector on page 133
- SailPoint IdentityIQ Siebel Connector on page 139
- SailPoint IdentityIQ Lotus Domino Connector on page 145
- SailPoint IdentityIQ Microsoft SQL Server on page 157
- SailPoint IdentityIQ Oracle Connector on page 163
- SailPoint IdentityIQ Sybase Connector on page 169
- SailPoint IdentityIQ Windows Local Connector on page 175
- SailPoint IdentityIQ AIX Connector on page 183
- SailPoint IdentityIQ Linux Connector on page 197
- SailPoint IdentityIQ Solaris Connector on page 207
- SailPoint IdentityIQ BMC Remedy IT Service Management Suite Connector on page 219
- SailPoint IdentityIQ Jive Connector on page 225
- SailPoint IdentityIQ Oracle E-Business Suite Connector on page 229
- SailPoint IdentityIQ Rally Connector on page 235
- SailPoint IdentityIQ BMC Remedy Connector on page 239
- SailPoint IdentityIQ RSA Ace Server Connector on page 245
- SailPoint IdentityIQ Salesforce/Remedyforce Connector on page 251
- SailPoint IdentityIQ SAP Connector on page 257
- SailPoint IdentityIQ SAP Enterprise Portal Connector on page 265
- SailPoint IdentityIQ Tivoli Access Manager Connector on page 271
- SailPoint IdentityIQ Tenrox Connector on page 277

The following note is applicable for all the Read/Write Direct Connectors mentioned above:

Note: The license for the SailPoint IdentityIQ Provisioning Engine Module is required before implementing the provisioning capabilities associated with the connector.

Chapter 1: SailPoint IdentityIQ Active Directory Connector

The following topics are discussed in this chapter:

| | |
|--|----|
| Overview | 27 |
| Supported features | 27 |
| Supported Managed System | 28 |
| Pre-requisites | 28 |
| Administrator permissions | 28 |
| Configuration parameters | 29 |
| Schema attributes | 30 |
| Account attributes | 30 |
| Group attributes | 35 |
| Provisioning Policy attributes | 36 |
| Install and register the IQService for Windows | 39 |
| Additional information | 40 |
| Unstructured Target Collector | 40 |

Overview

IdentityIQ mainly uses the LDAP and ADSI interfaces to Active Directory to communicate with Windows Domain Controllers. There are two types of group membership in Active Directory:

- primary group concept
- other group membership

In Active Directory you can only have one primary group, but any number of other groups. The other groups are listed as a property of the user object in Active Directory. When a user object is called it contains a list of groups in the member attribute. The primary group, however, is *not* listed as a group in the member attribute, so the connector must do a follow-up query to determine of which primary group the user is a member. The Active Directory connector uses the `primaryGroupSearchDN` attribute as the starting point when searching for a user's primary group.

Supported features

The Active Directory connector provides the ability to provision users, groups, and entitlements from IdentityIQ. The connector supports the following functions:

- Create/Update/Delete User
- Create/Update/Delete Group
- Manage Terminal Services, Dial-in Attributes
- Add custom attributes to provisioning policy to set the extended attributes
- Manage Exchange 2007, Exchange 2010, Exchange 2013
- Enable/Disable/Unlock/Reset Password for Users
- Add/Remove entitlements
- Pass through Authentication
- Password Interceptor

Overview

Password Interceptor for Active Directory provides the mechanism by which a password change initiated by an Active Directory user is captured by the Client and sent to IdentityIQ. For more information, see [Appendix A: Password Interceptor](#).

- Delta Aggregation
For more information, see [Appendix C: Delta Aggregation](#).
- Target Aggregation
For more information see, “[Unstructured Target Collector](#)” section.

Supported Managed System

- Supported Active Directory Domain Services (AD DS) functional levels
 - Microsoft Windows Server 2003
 - Microsoft Windows Server 2008
 - Microsoft Windows Server 2008 R2
 - Microsoft Windows Server 2012
- Supported Microsoft Exchange Servers
 - Microsoft Exchange Server 2007
 - Microsoft Exchange Server 2010
 - Microsoft Exchange Server 2013

Pre-requisites

1. Windows OS with Microsoft .NET Framework Version 2.0 or later.
2. Before you can use the provisioning feature of the connector, the IQService must be installed and registered. For more information about installing IQService see, “[Install and register the IQService for Windows](#)” on page 39.
3. For Exchange provisioning, it is required that the IQService host computer should be a Exchange Server or have Exchange Management Tools installed.
4. For managing Terminal Services (Remote Desktop Services profile) attributes, install the IQService on a Server class Windows Operating System.

Administrator permissions

For user provisioning through IQService, required that the administrator have the appropriate rights on the Active Directory. The Domain Controller should be accessible from the IQService host computer.

Note: The rights discussed in the following section grant limited account creation privileges to a user. This user can create and modify most accounts. It cannot manage the Administrator user account, the user accounts of administrators, the Server Operators, Account Operators, Backup Operators, and Print Operators. To manage these user types you must assign the appropriate security rights or add the user to groups having higher permissions. For example, domain administrators.

The administrative user specified in the application configuration will need additional rights for provisioning. These rights can be assigned by adding the user to the Account Operators group.

More granular rights can be assigned to users for specific portions of the directory, but this is discouraged by Microsoft best practices for Active Directory access control. The required rights will depend on the IdentityIQ use cases that are implemented, but could include

- Read All Properties
- Write All Properties
- Create User Objects
- Delete User Objects
- Change Password
- Reset Password
- Read Members
- Write Members

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Active Directory connector uses the following connection parameters:

| Parameters | Description |
|---------------------|--|
| useSSL | Specifies if the connection is over ssl. |
| authorizationType | Translates to the Context.SECURITY_AUTHENTICATION property in the api. The attribute value is one of the following strings: none , simple , strong . |
| user | The user (administrator) name used to make the connection. It should be in the domainName\UserName format. For more information about the rights required for administrator, see “Provisioning Policy attributes” on page 36 . |
| password | The password for the administrator account. |
| port | The port the server is listening through. |
| host | The host of the server. |
| searchScope | The depth to search the tree. OBJECT_SCOPE , ONELEVEL_SCOPE , and SUBTREE_SCOPE . |
| searchDN | The search starting point. This is a DN string. |
| iterateSearchFilter | An optional filter that can be added to the configuration to scope the objects returned when the iterateObjects method is called. |
| pageSize | The number of objects to get, per page, when iterating over large numbers of objects. The default is 100. |
| filterString | This setting can be used to filter object as they are returned for an underlying application. Derived attributes can also be included in the filter. |

Schema attributes

| Parameters | Description |
|-------------------------|--|
| primaryGroupSearchDN | Where to start in the tree when resolving a user's group membership. This is a DN String. |
| groupHierarchyAttribute | The name of the attribute from the GROUP schema that represents the groups to which this account group belongs. Specifying an attribute here enables the native account group hierarchy model to be displayed through IdentityIQ. The default value is memberOf . Clear this field if you choose not to use IdentityIQ's database hierarchy model. |
| IQService Host | Host name of the system where IQService is installed. |
| IQService Port | Port number used by the IQService. |
| Exchange Version | The version of the Exchange Server if it needs to be managed by the application. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Account attributes

| Name | Description |
|------------------|--|
| businessCategory | The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: "engineering", "finance", and "sales". |
| carLicense | This attribute type contains the license plate or vehicle registration number associated with the user. |
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute contains the distinguished name by which the user is known. |
| departmentNumber | This attribute contains a numerical designation for a department within your enterprise. |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

| Name | Description |
|--------------------------|---|
| destinationIndicator | <p>This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute.</p> <p>Examples: "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom.</p> <p>Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.</p> |
| displayName | This attribute contains the preferred name to be used for this person throughout the application. |
| employeeNumber | This attribute contains the numerical identification key for this person within your enterprise. |
| employeeType | This attribute contains a descriptive type for this user, for example, contractor, full time, or part time. |
| facsimileTelephoneNumber | This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute. |
| givenName | <p>This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute.</p> <p>Examples: "John", "Sue", and "David".</p> |
| homePhone | This attribute contains the employee's home phone number. |
| homePostalAddress | This attribute contains the employee's mailing address. |
| homeMDB | Exchange mailbox store DN. Required for mailbox creation. |
| initials | <p>This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute.</p> <p>Examples: "J. A." and "J"</p> |
| internationalISDNNumber | <p>This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute.</p> <p>Example: "0198 444 444".</p> |
| l | <p>This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute.</p> <p>Examples: "Austin", "Chicago", and "Brisbane".</p> |
| mail | This attribute type contains the RFC822 mailbox for the user. |
| manager | This attribute type contains the distinguished name of the manager to whom this person reports. |

Schema attributes

| Name | Description |
|----------------------------|---|
| mailNickname | Exchange Alias. |
| mobile | This attribute type contains the mobile telephone number of this person. |
| msExchHideFromAddressLists | Hide from Exchange address lists. |
| msNPAllowDialin | Indicates whether the account has permission to dial in to the RAS server. |
| msNPCallingStationID | If this property is enabled, the server verifies the caller's phone number. If the caller's phone number does not match the configured phone number, the connection attempt is denied. |
| msRADIUSCallbackNumber | The phone number that is used by the server is set by either the caller or the network administrator. If this property is enabled, the server calls the caller back during the connection process. |
| msRADIUSFramedRoute | Define a series of static IP routes that are added to the routing table of the server running the Routing and Remote Access service when a connection is made. |
| msRADIUSFramedIPAddress | Use this property to assign a specific IP address to a user when a connection is made. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| pager | This attribute type contains the telephone number of this persons pager. |
| physicalDeliveryOfficeName | This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E". |
| postOfficeBox | This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27". |
| postalAddress | This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA". |
| postalCode | This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA. |

| Name | Description |
|---------------------------|---|
| preferredDeliveryMethod | This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone". |
| preferredLanguage | This attribute type contains the preferred written or spoken language of this person. |
| registeredAddress | This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$XYZ Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA". |
| roomNumber | This attribute type contains the room or office number or this persons normal work location. |
| secretary | This attribute type contains the distinguished name of this persons secretary. |
| seeAlso | This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object "cn=Elvis Presley,ou=employee,o=XYZ\, Inc." is related to the role objects "cn=Bowling Team Captain,ou=sponsored activities,o=XYZ\, Inc." and "cn=Dart Team,ou=sponsored activities,o=XYZ\, Inc.". Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values. |
| sn | This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: "Smith". |
| st | This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. Example: "Texas". |
| street | This attribute type contains site information from a postal address (that is, the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: "15 Main St.". |
| telephoneNumber | This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. |
| teletexTerminalIdentifier | The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute. |
| telexNumber | This attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute |

Schema attributes

| Name | Description |
|------------------------------------|---|
| title | This attribute type contains the persons job title. Each title is one value of this multi-valued attribute. Examples: "Vice President", "Software Engineer", and "CEO". |
| uid | This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: "s9709015", "admin", and "Administrator". |
| objectClass | The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either "top" or "alias". |
| memberOf | This attribute type contains the account group membership for this person on the application. |
| objectSid | Windows Security Identifier |
| sAMAccountName | This attribute type contains the sAMAccountName for this user. |
| primaryGroupID | This attribute type contains the RID of the this users primary group. |
| primaryGroupDN | This attribute type contains the distinguished name of this users primary group. |
| TS_TerminalServicesProfilePath* | The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server. |
| TS_TerminalServicesHomeDrive* | The root drive for the user. |
| TS_TerminalServicesHomeDirectory* | The root directory for the user. |
| TS_TerminalServicesInitialProgram* | The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server. |
| TS_TerminalServicesWorkDirectory* | The working directory path for the user. |
| TS_EnableRemoteControl* | A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session. |
| TS_AllowLogon* | A value that specifies whether the user is allowed to log on to the RD Session Host server. |
| TS_BrokenConnectionAction* | A value that specifies the action to be taken when a Remote Desktop Services session limit is reached. |
| TS_ReconnectionAction* | A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed. |
| TS_ConnectClientDrivesAtLogon* | A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started. |

| Name | Description |
|----------------------------------|--|
| TS_ConnectClientPrintersAtLogon* | A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled. |
| TS_DefaultToMainPrinter* | A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled. |
| TS_MaxConnectionTime* | The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated. |
| TS_MaxDisconnectionTime* | The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated. |
| TS_MaxIdleTime* | The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated. ^a |

a. * - Attributes with asterik mark (*) are the Terminal Services/Remote Desktop Services attributes. By default these attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Group attributes

Table 2—Active Directory Connector - Group Attributes

| Name | Description |
|-------|--|
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute type contains the directory path to the object. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| owner | This attribute type contains the name of the owner of the object. |

Provisioning Policy attributes

Table 2—Active Directory Connector - Group Attributes

| Name | Description |
|----------------|---|
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |
| mailNickname | Exchange Alias. |
| memberOf | This attribute type contains the account group membership for this person on the application. |
| objectSid | This attribute type contains the Windows Security Identifier for this user. |
| sAMAccountName | sAMAccountName |
| groupType | Group Type. Allowed values are: 1. Security 2. Distribution |
| groupScope | Group Scope. Allowed values are: 1. Domain local 2. Global 3. Universal. |

Provisioning Policy attributes

The following table lists the provisioning policy attributes:

| Attribute | Description |
|--|--|
| Provisioning policy attributes for Account Creation | |
| ObjectType | Type of the user to be created. Default value: User. |
| distinguishedName | Distinguished name of the user to be created. |
| sAMAccountName | sAMAccountName of the user to be created. |
| *password* | Password of the user to be created. |
| IIQDisabled | A boolean attribute, set to true to create a disabled user. |
| PrimaryGroupDN | Default group of the user to be created. |
| description | Description of the user to be created. |
| msNPAllowDialin | Indicates whether the account has permission to dial in to the RAS server. |

| Attribute | Description |
|------------------------------------|--|
| msNPCallingStationID | If this property is enabled, the server verifies the caller's phone number. If the caller's phone number does not match the configured phone number, the connection attempt is denied. |
| msRADIUSCallbackNumber | The phone number that is used by the server is set by either the caller or the network administrator. If this property is enabled, the server calls the caller back during the connection process. |
| msRADIUSFramedRoute | Define a series of static IP routes that are added to the routing table of the server running the Routing and Remote Access service when a connection is made. |
| msRADIUSFramedIPAddress | Use this property to assign a specific IP address to a user when a connection is made. |
| TS_TerminalServicesProfilePath* | The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server. |
| TS_TerminalServicesHomeDrive* | The root drive for the user. |
| TS_TerminalServicesHomeDirectory* | The root directory for the user. |
| TS_TerminalServicesInitialProgram* | The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server. |
| TS_TerminalServicesWorkDirectory* | The working directory path for the user. |
| TS_EnableRemoteControl* | A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session. |
| TS_AllowLogon* | A value that specifies whether the user is allowed to log on to the RD Session Host server. |
| TS_BrokenConnectionAction* | A value that specifies the action to be taken when a Remote Desktop Services session limit is reached. |
| TS_ReconnectionAction* | A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed. |
| TS_ConnectClientDrivesAtLogon* | A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started. |
| TS_ConnectClientPrintersAtLogon* | A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled. |
| TS_DefaultToMainPrinter* | A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled. |

Provisioning Policy attributes

| Attribute | Description |
|---|--|
| TS_MaxConnectionTime* | The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated. |
| TS_MaxDisconnectionTime* | The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated. |
| TS_MaxIdleTime* | The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated. ^a |
| Special provisioning attributes for Move/Rename request | |
| AC_NewName | A string attribute to rename the user. For example, CN=abc |
| AC_NewParent | A string attribute to move the user to new OU. For example, OU=xyz,DC=pqr,DC=com |
| Provisioning Exchange mailbox | |
| homeMDB | Exchange mailbox store DN. Required for mailbox creation. Optional for Exchange 2010 and above. |
| mailNickname | Exchange alias. Required for mailbox creation and to update or disable the mailbox. Send this attribute with no value to disable the mailbox. |
| msExchHideFromAddressLists | (<i>Optional</i>) Hide from Exchange address lists. |
| DomainController | (<i>Optional</i>) Fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. |
| <p>Note: Connector supports updating any other Exchange mailbox attributes supported by set-mailbox cmdlet. To set any such parameter, prefix the parameter name of the set-mailbox cmdlet with Exch_ while adding the attribute to the provisioning policy.</p> <p>To enable provisioning Exchange 2013, rename app.config file located at IQService home to IQService.exe.config and restart the IQService.</p> | |
| Provisioning policy attributes for CreateGroup | |
| distinguishedName | Group in the distinguished name format. |
| sAMAccountName | sAMAccountName |
| Provisioning policy attributes for UpdateGroup | |
| description | A description of the group. |

| Attribute | Description |
|--------------|--|
| groupType | Group Type. Allowed values are: 1. Security 2. Distribution |
| groupScope | Group Scope. Allowed values are: 1. Domain local 2. Global 3. Universal. |
| mailNickname | Alias and is required if want to create Distribution Group on exchange. Only Universal type of group can be created on exchange. |

a. * - Attributes with asterik mark (*) are the Terminal Services/Remote Desktop Services attributes. By default these attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Install and register the IQService for Windows

You must install and register an IQService before you can provision to Active Directory, aggregate Terminal Services attributes, collect information from the Windows Event Logs, or load local Windows users or groups. The IQService is a native Windows service that enables IdentityIQ to participate in a Windows environment and access information only available through Windows APIs.

To install and register the IQService, perform the following:

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the **IQService.zip** archive from the **IIQHOME\WEB-INF\bin\win** directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.
IQService.exe -i
4. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s

Additional information

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information

Note: To enable provisioning Microsoft Exchange 2013, rename app.config file located at IQService home to IQService.exe.config file.

Additional information

This section describes the additional information related to the Active Directory Connector.

Unstructured Target Collector

Unstructured target information is used to define unstructured data sources from which the connector is to extract data. Unstructured data is any data that is stored in a format that is not easily readable by a machine. For example, information contained in an Excel spread sheet, the body of an email, a Microsoft Word document, or an HTML file is considered unstructured data. Unstructured targets pose a number of challenges for IdentityIQ connectors, because not only is the data stored in a format that is hard to extract from, the systems and directory structures in which the files reside are often difficult to access.

The most common unstructured data type supported by IdentityIQ is an operating system's file system permissions.

This target collector requires a the IdentityIQ Service to be installed on a machine that has visibility to the directory or share to include in the target scan. Refer to the Installation Guide for information on installing and registering the IQService.

The unstructured targets defined on this tab are used by the Target Aggregation task to correlate targets with permissions assigned to identities and account groups for use in certifications.

The Unstructured Targets tab contains the following information:

Table 3—Application Configuration - Unstructured Targets Tab field descriptions

| Field | Description |
|---|--|
| Attributes: The required settings for connecting to the IdentityIQ Service. | |
| IQService Host | The host on which the IdentityIQ Service resides. |
| IQService Port | The TCP/IP port where the IQService is listening for requests. |
| Number of targets per block | Number or targets (files) to include in each block of data returned. |
| File Shares: The required information for each share. | |

Table 3—Application Configuration - Unstructured Targets Tab field descriptions

| Field | Description |
|---|---|
| Path | UNC Style path to a share or local directory. You can target a specific file or a directory and its sub-directories containing multiple files from which to extract the required data. If you target a directory, use the Wildcard and Directory Depth fields to narrow the query if possible. |
| Directories Only | Use to instruct the collector to ignore files and just report back directory permission information. |
| Directory Depth | The sub-directory depth from which to extract data. The Directory Depth field enables you to extend your query up to ten (10) sub-directories below the one specified in the Path field. |
| Wildcard | Use wild cards to target a particular file type or naming scheme. For example, to search only Excel spread sheets, use *.xls or to search only files with names beginning with finance_ , use finance_* . |
| Include Inherited Permissions | Use to instruct the collector to not report permissions unless they are directly assigned. Only directly assigned permissions will be returned |
| Administrator | The administrator that has access to this share so you can collect permissions. This value should be the users principal user@xyz.com name or a fully qualified domain user name in the domain\user format. |
| Password | The password associated with the specified administrator. Note: The service will be running as System or can be configured to be run as any user, so the Administrator/Password fields may not be required in all cases. |
| Rules: Specify the rules used to transform and correlate the targets. Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed. | |
| Creation Rule | The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ. |
| Correlation Rule | The rule used to determine how to correlate account information from the application with identity cubes in IdentityIQ. |

Additional information

Chapter 2: SailPoint IdentityIQ WebEx Connector

The following topics are discussed in this chapter:

| | |
|--|----|
| Overview | 43 |
| Supported features | 43 |
| Administrator permissions | 43 |
| Configuration parameters | 43 |
| Schema attributes | 44 |
| Account attributes | 44 |
| Group attributes | 46 |
| Provisioning Policy attributes | 46 |

Overview

The WebEx connector manages WebEx accounts and groups (Meeting Types). It supports read and write for WebEx accounts. The WebEx connector supports creation, deletion, retrieval, authentication and unlock for users and retrieval for groups.

Note: In the WebEx connector, Meeting Types are treated as Groups.

Supported features

This release of the connector provides support for the following operations:

- Account Aggregation
- Account-Group Aggregation
- Request entitlement
- Create/Delete/Refresh user
- Enable/Disable user
- Lock/Unlock user
- Managed password

Administrator permissions

The user must be a **Site Administrator**.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The WebEx Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|------------|-------------------------------------|
| webExID | WebEx user ID for the meeting host. |

Schema attributes

| Parameters | Description |
|--------------------------|---|
| password | The password for the user with a webExID. |
| siteID | The WebEx-assigned identification number that uniquely identifies your website. |
| siteName | The first string in your WebEx site URL, provided by WebEx. For example, is acme is the siteName for the https://acme.webex.com site. |
| partnerID | (Optional) A reference to the WebEx partner, provided by WebEx. |
| xmlURL | XML URL of the site. For example, WBXService/XMLService |
| Manage Disabled Accounts | If set to yes, the disabled accounts will be a part of the Aggregation. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|------------------|---|
| WebexID | WebEx ID of the user. |
| FirstName | First Name of the user. |
| LastName | Last Name of the user. |
| Email | Email id of the user. |
| RegistrationDate | The creation date of the user |
| Active | Determines whether the user account has been staged for use. Default: ACTIVATED Note: If you set the user's active parameter to ACTIVATED such that the WebEx sites host limit is exceeded, the CreateUser or SetUser command displays the following error: exceededSiteHostLimit |
| TimezoneID | Determines the time zone for the geographic location of the meeting. |
| Company | The user's company name. |

| Attributes | Description |
|---|--|
| Description | A description of the user's virtual office. |
| CategoryID | A reference to the office category for the user's office. |
| AddressType | Determines whether the meeting participant is a personal contact of the meeting host or is a site-wide (global) contact. |
| Country | The country for the user. |
| Phone | The user's Office Profile phone number. |
| MobilePhone | The attendee's mobile phone number. |
| Fax | Indicates the fax number for the user. |
| Pager | The user's Office Profile pager number. |
| PersonalURL | The user's website. |
| ExpirationDate | A WebEx-maintained date and time at which the user's account expires. |
| Prod/ServiceAnnouncement | Indicates product or service announcements. |
| TrainingInfo | Indicates training information. |
| ElectronicInfo | Indicates electronic information. |
| Promos | Indicates promotions and special offers. |
| PressRelease | Indicates press releases. |
| UserEmail | The email address as stored in the user profile. |
| UserPhone | Indicates the phone number for the user. |
| MailInfo | Indicates the mail information for the user. |
| TimeZone | Determines the time zone for the geographic location of the user or user's office. |
| TimeZoneWithDST | A timezone description which is adjusted by DST. For example, GMT-7:00, Pacific (San Francisco) |
| Service | The type of service that the user has. |
| Host | Indicates whether the user is the host for the meeting. |
| TelephoneConferenceCallOut | Indicates whether conference calling out of meetings is supported for the meeting. |
| TelephoneConferenceCallOutInternational | Indicates whether international calling out of meetings is supported for the meeting. |
| TelephoneConferenceCallIn | Indicates whether conference calling into meetings is supported for the meeting. |
| TelephoneConferenceTollFreeCallIn | Indicates whether toll-free calling into meetings is supported for the user. |
| SiteAdmin | Indicates whether the user has administrative privilege for the meeting. |
| VOIP | Specifies whether Voice Over IP telephony is enabled. |

Provisioning Policy attributes

| Attributes | Description |
|-----------------------------------|--|
| SiteAdminwithViewOnly | Indicates whether the current user is a site administrator with view only privilege. |
| LabAdmin | If TRUE, then user has access to the Hands-on Lab administration pages. |
| OtherTeleConferencing | Specifies whether a user account has the privilege to schedule a session with the other teleconferencing feature enabled. Default value depends on the configurations on the user's website. |
| TeleConferenceCallInInternational | Allows a user to access WebEx teleconferencing through international local call-in telephone numbers. |
| AttendeeOnly | If the value is TRUE, indicates that the user's role is attendee only. If the value is set to TRUE, then the host , siteAdmin , labAdmin and roSiteAdmin elements should be FALSE. |
| RecordingEditor | Indicates whether a user has the privilege to download WebEx Recording Editor from My WebEx > Support. |
| MeetingAssist | Enables Meeting Assist. |
| MeetingType | The meeting types of which the account is a part of. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|-----------------------------|---|
| ProductCodePrefix | Indicates the product label for the type of meeting. |
| Active | Indicates whether the type of meeting represented by an object of this type is enabled or disabled. |
| DisplayName | The display name for the meeting type. |
| PrimaryTollCallInNumber | The telephone number for a toll call-in teleconference. |
| PrimaryTollFreeCallInNumber | The telephone number for a toll free call-in teleconference. |
| GroupName | The name of the group or meeting type |
| MeetingTypeID | Specifies IDs for the meeting types whose detailed information you want to get. |
| ServiceType | The type of meeting being returned. |

Provisioning Policy attributes

The following table lists the provisioning policy attributes for create:

| Attributes | Description |
|---|---|
| AccountType | Host: Indicates whether the user is the host for the meeting. |
| | SiteAdmin: Indicates whether the user has administrative privilege for the meeting. |
| | SiteAdminWithViewOnly: Indicates whether the current user is a site administrator with view only privilege. |
| WebexID | WebEx ID of the user. |
| FirstName | First Name of the user. |
| LastName | Last Name of the user. |
| Email | Email ID of the user. |
| TelephoneConferenceCallOut | Indicates whether conference calling out of meetings is supported for the meeting. |
| TelephoneConferenceCallOutInternational | Indicates whether international calling out of meetings is supported for the meeting. |
| TelephoneConferenceCallIn | Indicates whether conference calling into meetings is supported for the meeting. |
| TelephoneConferenceTollFreeCallIn | Indicates whether toll-free calling into meetings is supported for the user. |
| VOIP | Specifies whether Voice Over IP telephony is enabled. |
| LabAdmin | If TRUE, then user has access to the Hands-on Lab administration pages. |
| OtherTeleConferencing | Specifies whether a user account has the privilege to schedule a session with other teleconferencing feature enabled. Default value depends on the configurations on the user's website. |
| TeleConferenceCallInInternational | Allows a user to access WebEx teleconferencing via international local call-in telephone numbers. |
| RecordingEditor | Indicates whether a user has the privilege to download WebEx Recording Editor from My WebEx > Support. |
| WelcomeMessage | Holds a welcome message for when people enter the meeting room. |

Provisioning Policy attributes

Chapter 3: SailPoint IdentityIQ Google Apps Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|----|
| Overview | 49 |
| Supported features | 49 |
| Pre-requisites | 49 |
| Administrator permissions | 49 |
| Configuration parameters | 50 |
| Schema attributes | 50 |
| Account attributes | 50 |
| Group attributes | 50 |
| Provisioning Policy attributes | 51 |

Overview

SailPoint Google Apps connector manages Google Apps users and groups.

Supported features

The connector supports the following functions:

- Account Aggregation
- Account-Group Aggregation
- Create\Delete\Refresh Account
- Create\Update\Delete Account-Group
- Add\Remove Entitlement
- Enable\Disable Account
- Change Password
- Authenticate

Pre-requisites

- Administrator username and password for your Google Apps domain.
- Enable the Provisioning API for your Google Apps domain.

To enable the API, perform the following steps:

1. Log in to your administrator account.
2. Click **Domain settings => User settings** and select the **Provisioning API** check box.
3. Save the changes.

Administrator permissions

The Administrator should be configured to have proper access rights for modifying Google Apps users.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Google Apps Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|---------------|---|
| Domain | The name of the Google Apps registered domain that must be managed. |
| Administrator | The Google Apps user with administrative rights. All the requests are executed with this users context. |
| Password | Password of the administrative user. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|------------|--|
| UserID | User ID of the specified user. |
| GivenName | Given name of the specified user. |
| FamilyName | Family name of the specified user. |
| Email | Email of the specified user. |
| Nickname | Nick name of the specified user. |
| Groups | Groups to which the specified user is connected. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|------------|--------------------------------|
| GroupName | Name of the specified group. |
| GroupEmail | E-mail of the specified group. |

| Attributes | Description |
|------------------|-------------------------------------|
| GroupDescription | Description of the specified group. |
| GroupOwner | Owners of the specified group. |
| GroupPermission | Permissions of the specified group. |
| GroupRole | Roles of the specified group. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes:

| Attributes | Description |
|--|---|
| Provisioning policy attributes for Account Creation | |
| UserID | User ID of the user to be created. |
| GivenName | Given name of the user to be created. |
| FamilyName | Family name of the user to be created. |
| Password | Password of the user to be created. |
| AdminPrivileges | Whether the user to be created should have administrative privileges. |
| quota(optional) | Maximum email quota allowed for the user to be created. |
| Provisioning policy attributes for Group Creation | |
| GroupEmail | E-mail of the specified group. |
| Provisioning policy attributes for Group Update | |
| GroupName | Name of the specified group. |
| GroupDescription | Description of the specified group. |

Provisioning Policy attributes

Provisioning Policy attributes

Chapter 4: SailPoint IdentityIQ LDAP Connector

The following topics are discussed in this chapter:

| | |
|---|----|
| Overview | 55 |
| Supported features | 55 |
| Supported Managed Systems | 56 |
| Pre-requisites | 57 |
| Administrator permissions | 58 |
| Configuration parameters | 58 |
| Schema attributes | 59 |
| Account attributes | 59 |
| Group attributes | 62 |
| Group Membership attribute | 63 |
| Group Entitlement attribute | 63 |
| Additional information | 64 |
| Support for NISNetGroups and POSIXGroups | 64 |
| Managing Revoke-Restore for SunOne | 65 |
| Using Novell eDirectory as a Pass-through Authentication Source | 66 |

Overview

This connector was developed using the LDAP RFC. The LDAP connector should plug into almost any LDAP server with no customization. The LDAP Connector now supports provisioning of users and entitlements along with the retrieval of LDAP account and group object classes.

Supported features

SailPoint IdentityIQ LDAP Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Create/Update/Delete Account
- Create/Update/Delete Account-Group
- Account Refresh
- Add/Remove Entitlement
- Enable/Disable/Unlock Account
- Change/Reset password
- Pass through Authentication
- Password Interceptor

Password Interceptor for LDAP provides the mechanism by which a password change initiated from LDAP system is captured by the Client and sent to IdentityIQ. For more information, see [“Password Interceptor for LDAP” on page 358](#).

- Delta Aggregation

For more information, see [Appendix , “C: Delta Aggregation”](#).

Overview

- NISNetGroups and POSIXGroups

For more information, see [“Support for NISNetGroups and POSIXGroups”](#) section.

Supported Managed Systems

SailPoint IdentityIQ LDAP connector supports the following Managed Systems:

- Microsoft ADAM version 1.0, 1.0.230.0
- OpenLDAP version 2.3.39, 2.4.26
- SunOne Directory Server version 5.2, 6.3
- IBM Tivoli Directory Server version 6.0, 6.1
- Novell eDirectory version 8.8.5
- Oracle Internet Directory version 10gR3
- Oracle Internet Directory version 11gR2

For each of the supported managed systems we have application types already existing in the connector registry as mentioned in the following section.

LDAP Connector Application Types

In order to speed up the process of building an application quickly, IdentityIQ comes with the following default LDAP application types to manage the corresponding directory server:

| Application Types | Description |
|------------------------------------|---|
| ADAM - Direct | Manages ADAM directory server. |
| SunOne - Direct | Manages SunOne directory server. |
| IBM Tivoli DS - Direct | Manages Tivoli directory server. |
| Novell eDirectory - Direct | Manages Novell eDirectory server. |
| Oracle Internet Directory - Direct | Manages Oracle Internet directory server. |
| OpenLDAP - Direct | Manages OpenLDAP directory servers. |

Note: The read-only LDAP connector available prior to version 5.5 patch 1 with application type LDAP is still available and will continue to function exactly as before.

The following table displays the object classes mapped for each of the LDAP application types:

Table 4—Object classes mapped for each of the LDAP application types

| Appication type | Objectclass mapped for accounts | Objectclass mapped for groups | Group member-ship attributes | Group entitle-ment attribute |
|----------------------------|---------------------------------|-------------------------------|------------------------------|------------------------------|
| ADAM - Direct | user | group | member | groups |
| SunOne - Direct | inetOrgPerson | groupofUniqueNames | uniquemember | groups |
| IBM Tivoli DS - Direct | inetOrgPerson | groupofUniqueNames | uniqueMember | groups |
| Novell eDirectory - Direct | inetOrgPerson | groupofUniqueNames | uniqueMember | groups |

Table 4—Object classes mapped for each of the LDAP application types

| Application type | Objectclass mapped for accounts | Objectclass mapped for groups | Group membership attributes | Group entitlement attribute |
|------------------------------------|---------------------------------|-------------------------------|-----------------------------|-----------------------------|
| Oracle Internet Directory - Direct | inetOrgPerson | groupofUniqueNames | uniqueMember | groups |
| OpenLDAP - Direct | inetOrgPerson | groupofUniqueNames | uniqueMember | groups |

Create SSL Communication Between IdentityIQ and ADAM Server

If you want secure SSL connection for ADAM, SSL communication needs to be enabled between IdentityIQ and ADAM Server. For a Java client to connect using SSL and self-signed certificates, you have to install the certificate into the JVM keystore.

To create SSL communication between IdentityIQ and ADAM Server, perform the following:

1. Export server certificate and copy the exported .cer file to the Java client computer (IdentityIQ computer).
2. At the client computer execute the following command from the bin directory of JDK:
keytool -importcerts -trustcacerts -alias aliasName -file <absolute path of certificate> -keystore <JAVA_HOME>/jre/lib/security/cacerts
 In the preceding command line, *aliasName* is the name of the alias and *jksFileName* is the name of the jks file.
3. Login to IdentityIQ.
4. Create the application for ADAM Direct, use SSL and provide all the required values.
5. Click on **Test Connection** and save the application.

Object Types Managed

Each of the application types has been preconfigured to manage commonly used object classes and their attributes. For instance, the application schema of ADAM directory server has been configured for user and group object classes. [Table 4—Object classes mapped for each of the LDAP application types](#) shows the mapping for various application types. Custom object classes may be mapped by modifying the corresponding application schema.

Connector Type Names for Provisioning Module for LDAP

When upgrading IdentityIQ to version 6.1, the names of the LDAP Gateway Connectors have been changed to the following:

- **ADAM - Gateway** has been changed to **ADAM - Full**
- **SunOne - Gateway** has been changed to **SunOne - Full**
- **Tivoli - Gateway** has been changed to **Tivoli - Full**

Note: For a fresh installation of IdentityIQ version 6.1, the ADAM- Gateway, SunOne – Gateway and Tivoli – Gateway are not available in the list of supported application types.

Pre-requisites

SailPoint IdentityIQ LDAP Connector requires that the directory server has the administrator credentials.

Administrator permissions

SailPoint IdentityIQ LDAPConnector must have the read /write privilege's over the directory information tree in order to manage the LDAP data.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The LDAP connector uses the following connection attributes:

| Parameters | Description |
|-------------------------|--|
| useSSL | Specifies if the connection is over ssl. |
| authorizationType | Translates to the Context.SECURITY_AUTHENTICATION property in the api. The attribute value is one of the following strings: none , simple , strong . |
| user | The user to connect as. Typically a DN string such as Administrator. |
| password | The password for the administrator account. |
| port | The port the server is listening through. |
| host | The host of the LDAP server. |
| searchScope | The depth to search the LDAP tree. OBJECT_SCOPE , ONELEVEL_SCOPE , and SUBTREE_SCOPE . |
| searchDN | The search starting point. This is a DN string. |
| iterateSearchFilter | An optional filter that can be added to the configuration to scope the objects returned when the iterateObjects method is called. |
| pageSize | The number of objects to get, per page, when iterating over large numbers of objects. The default is 500. |
| groupMemberSearchDN | Where to start in the tree when resolving a user's group membership. This should contain the DN(s) of one or more container(s) delimited by a semicolon. |
| filterString | This setting can be used to filter object as they are returned for an underlying application. Derived attributes can also be included in the filter. |
| groupMemberAttribute | The name of the attribute used on the LDAP server to store group members. |
| Enable Net Groups | <i>(Applicable only for IBM Tivoli DS - Direct, OpenLDAP - Direct, SunOne - Direct)</i> Enable this check-box to retrieve NisNetGroups entitlements. |
| map to member attribute | <i>(Applicable only for IBM Tivoli DS - Direct, OpenLDAP - Direct, SunOne - Direct)</i> The user attribute specifying user membership in nisnetgroup . Example uid attribute of the user. |
| Enable Posix Groups | <i>(Applicable only for IBM Tivoli DS - Direct, OpenLDAP - Direct, SunOne - Direct)</i> Enable this check-box to retrieve PosixGroups entitlements. |

| Parameters | Description |
|-------------------------|---|
| map to member attribute | <i>(Applicable only for IBM Tivoli DS - Direct, OpenLDAP - Direct, SunOne - Direct)</i> The user attribute specifying user membership in posixgroup . Example uid attribute of the user. |

Note: The LDAP connector uses the **groupMemberSearchDN** attribute as the starting point in the directory to start searching for ALL group memberships. LDAP does not store a user's group references on the user so the LDAP connector must always do a separate query to return a list of all of the user's groups.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 5—LDAP Connector - Account Attributes

| Name | Description |
|------------------|--|
| businessCategory | The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: "engineering", "finance", and "sales". |
| carLicense | This attribute type contains the license plate or vehicle registration number associated with the user. |
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute contains the distinguished name by which the user is known. |
| departmentNumber | This attribute contains a numerical designation for a department within your enterprise. |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Table 5—LDAP Connector - Account Attributes (Continued)

| Name | Description |
|--------------------------|--|
| destinationIndicator | <p>This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom.</p> <p>Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.</p> |
| displayName | This attribute contains the preferred name to be used for this person throughout the application. |
| employeeNumber | This attribute contains the numerical identification key for this person within your enterprise. |
| employeeType | This attribute contains a descriptive type for this user, for example, contractor, full time, or part time. |
| facsimileTelephoneNumber | This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute. |
| givenName | <p>This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. Examples: "John", "Sue", and "David".</p> |
| groups | <p>This attribute type contains a list of groups of which this person is a member. Example: "Sales" or "Engineering"</p> |
| homePhone | This attribute contains the employee's home phone number. |
| homePostalAddress | This attribute contains the employee's mailing address. |
| initials | <p>This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute. Examples: "J. A." and "J".</p> |
| internationalISDNNumber | <p>This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. Example: "0198 444 444".</p> |
| l | <p>This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. Examples: "Austin", "Chicago", and "Brisbane".</p> |
| mail | This attribute type contains the RFC822 mailbox for the user. |

Table 5—LDAP Connector - Account Attributes (Continued)

| Name | Description |
|----------------------------|---|
| manager | This attribute type contains the distinguished name of the manager to whom this person reports. |
| mobile | This attribute type contains the mobile telephone number of this person. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| pager | This attribute type contains the telephone number of this persons pager. |
| physicalDeliveryOfficeName | This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E". |
| postOfficeBox | This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27". |
| postalAddress | This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA". |
| postalCode | This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA. |
| preferredDeliveryMethod | This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone". |
| preferredLanguage | This attribute type contains the preferred written or spoken language of this person. |
| registeredAddress | This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA". |
| roomNumber | This attribute type contains the room or office number or this persons normal work location. |
| secretary | This attribute type contains the distinguished name of this persons secretary. |

Table 5—LDAP Connector - Account Attributes (Continued)

| Name | Description |
|---------------------------|---|
| seeAlso | This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object “cn=Elvis Presley,ou=employee,o=xyz\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.”. Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values. |
| sn | This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: “Smith”. |
| st | This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”. |
| street | This attribute type contains site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”. |
| telephoneNumber | This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. |
| teletexTerminalIdentifier | The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute. |
| telexNumber | This attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute |
| title | This attribute type contains the persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”. |
| uid | This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”. |
| objectClass | The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 6—LDAP Connector - Group Attributes

| Name | Description |
|--------------|--|
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| uniqueMember | This attribute type contains the groups to which this person is a unique member. |
| dn | This attribute type contains the directory path to the object. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| owner | This attribute type contains the distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is "cn=All Employees, ou=Mailing List,o=xyz, Inc.", is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): "cn=Human Resources Director,ou=employee,o=xyz, Inc." |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Group Membership attribute

The group membership attribute has been implicitly mapped for the various application types. This attribute and its value can be seen in the application page. Refer to [Table 4—Object classes mapped for each of the LDAP application types](#) for the group membership attribute mapped for each application type. This attribute can be changed from the default to a membership attribute specific to the custom object class mapped. For instance, if the groupOfUniqueNames is the default object class that has been mapped in the application schema for managing groups, then the default group membership attribute can be changed from uniquemember to member if groupOfNames is mapped in the application schema to manage groups.

Group Entitlement attribute

By default, all application types have the groups attribute mapped as the default entitlement attribute. This attribute is simply a placeholder to contain user/group memberships. While creating a new group aggregation task for the application, you would need to specify the value groups in the group account attribute text box in the group aggregation task.

Additional information

This section describes the additional information related to the LDAP Connector.

Support for NISNetGroups and POSIXGroups

This feature retrieves and manages NISNetGroup membership and POSIXGroup membership of an LDAP user from IdentityIQ. This feature does not require to create a new schema with nisnetgroup or posixgroup objectclass explicitly mapped as **nativeobjecttypes**, the feature has been implemented with nisnetgroup and posixgroup mapped as entitlement attributes.

The NISNetGroups and POSIXGroups support has been implemented for and tested on the following application types:

- SunOne - Direct
- IBM Tivoli DS - Direct
- OpenLDAP - Direct

It can be configured for fresh and existing applications. Perform the following procedure for configuring the NISNetGroups and POSIXGroups support:

1. Perform one of the following:

- To support NisNetGroups:
 - a. In the application configuration page, select the **Enable Net Groups** check-box and enter appropriate value for the **Map To Member Attribute** field.
 - b. Add the nisnetgroups account schema attributes with the managed, entitlement and multivalued checkboxes selected.
 - c. By default, LDAP Connector supports rounded braces in the **nisNetGroupTriple** membership value. Other brace types in the **nisNetGroupTriple** value may be supported using the **nisNetGroupTriple_Brace_Override** configuration attribute. For example, to specify curly braces in the **nisNetGroupTriple** configuration attribute, add the following entry in the application debug page:

```
<entry key="nisNetGroupTriple_Brace_Override" value="{"/>
```

Other supported brace types are angular and square.

Note: To specify angular braces, add the following entry in the application debug page:

```
<entry key="nisNetGroupTriple_Brace_Override" value="&lt;"/>
```

Note: For an application, LDAP Connector supports only one brace type.

Note: For a given server instance, if nisNetGroupTriple attribute has been populated with a certain brace type, then the same brace type has to be specified in the LDAP Connector application configuration debug page.

- To support PosixGroups:
 - a. In the application configuration page, select the **Enable Posix Groups** check-box and enter appropriate value for the **Map To Member Attribute** field.
 - b. Add the posixgroups account schema attributes with the managed, entitlement and multivalued checkboxes selected.
2. This feature does not require to create a new schema with nisnetgroup or posixgroup objectclass explicitly mapped as nativeobjecttypes, the feature has been implemented with nisnetgroup and posixgroup

mapped as entitlement attributes. Add the **nisnetgroups** and **posixgroups** account schema attributes with the **managed**, **entitlement** and **multivalued** checkboxes selected.

Note the following:

- For SunOne, only the nisnetgroup memberships with curly braces in **nisnetgrouptriple** attribute would be retrieved.

For example, {host1,user1,} is a supported format. Whereas, (host1,user1,) , [host1,user1,] , <host1,user1,> are unsupported formats. For instance, if user1 has a nisnetgroup memberships which is in the format any other than the curly braces, then this entitlement would not be retrieved into IdentityIQ.

- The NISNetGroup entitlement is added only with the user portion of the **nisnetgrouptriple** attribute value. The domain and host counterpart are not incorporated.

For example, on SunOne and Tivoli {,user1,} is the value of the **nisnetgrouptriple** attribute after adding an entitlement for user1 on a nisnetgroup.

- Latest version of OpenLDAP server on windows (2.4.26) is unable to perform an equality search on the **nisnetgrouptriple** attribute of **nisnetgroup** objectclass. As a result, the nisnetgroup membership is not displayed in IdentityIQ after a full aggregation.

To resolve this, open **nis.schema** file of the OpenLDAP server installation and verify if the **nisnetgrouptriple** schema attribute definition is same as the following attribute type:

```
(1.3.6.1.1.1.1.14 NAME 'nisNetgroupTriple'
DESC 'Netgroup triple'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
```

If the attribute type is not same as the above, then take a back up of the existing **nis.schema** file and replace the existing **nisnetgrouptriple** definition with the above. Save the file and restart the OpenLDAP server. After performing aggregation, nisnetgroup membership should now get fetched into IdentityIQ.

- The IdentityIQ 6.0 patch 5 incorporated the support of **POSIXACCOUNT** and **POSIXGROUP** feature. According to which, **posixaccount** and **posixgroup** were mapped as native object types in the application schema. After upgrading to IdentityIQ 6.1, the functionality would not work for the existing applications.

To ensure that the functionality works, in the application configuration page, select the **Enable Posix Groups** check-box and enter appropriate value for the **Map To Member Attribute** field.

For instance, if the configuration attribute appeared as

```
<entry key="groupMembershipAttributeType" value="uid"/>
```

then it should be changed to

```
<entry key="posixGroup_Member_Attribute" value="uid"/>
```

- The IdentityIQ 6.0 patch 6 incorporated the support of NetGroups and PosixGroups feature. If you have already configured application for supporting NetGroups and PosixGroups, after upgrading to 6.1, then perform one of the following:

- To support NisNetGroups: In the application configuration page, select the **Enable Net Groups** check-box and enter appropriate value for the **Map To Member Attribute** field.
- To support PosixGroups: In the application configuration page, select the **Enable Posix Groups** check-box and enter appropriate value for the **Map To Member Attribute** field.

Managing Revoke-Restore for SunOne

SunOne directory server requires the complete DN of the nsmanagedDisabledRole object to manage revoke-restore functionality.

Additional information

By default, in the application schema for SunOne - Direct, `nsmanagedDisabledRole` attribute has been mapped as follows to manage restore and revoke respectively:

```
<entry key="restoreVal" value="cn=nsManagedDisabledRole,dc=Naming Context"/>
```

and

```
<entry key="revokeVal" value="cn=nsManagedDisabledRole,dc=Naming Context"/>
```

You need to modify this attribute to contain the complete DN of the `nsManagedDisabledRole` object. For instance, if the DN of the `nsManagedDisabledRole` is `cn=nsManagedDisabledRole, dc=sailpoint, dc=com`, the restore entry would be modified to as follows:

```
<entry key="restoreVal" value="cn=nsManagedDisabledRole,dc=sailpoint,dc=com"/>
```

Similarly, you would need to modify the revoke xml entry as follows:

```
<entry key="revokeVal" value="cn=nsManagedDisabledRole,dc=sailpoint,dc=com"/>
```

Using Novell eDirectory as a Pass-through Authentication Source

If using the Novell eDirectory - Direct application type as a pass-through authentication source, remove the `dn` entry from the Authentication Search Attributes. Using the DN is currently not supported.

Chapter 5: SailPoint IdentityIQ Box.Net Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|----|
| Overview | 67 |
| Supported features | 67 |
| Supported Managed Systems | 67 |
| Pre-requisites | 67 |
| Administrator permissions | 68 |
| Configuration parameter | 68 |
| Schema attributes | 68 |
| Account attributes | 68 |
| Group attributes | 68 |
| Provisioning Policy attributes | 69 |

Overview

The Box.Net Connector manages managed users and groups of Box server. Box.Net connector is a read/write connector that can retrieve managed users and groups of a particular network, activates/inactivates managed users and assign managed users to groups.

Supported features

SailPoint IdentityIQ Box.Net Connector provides support for the following features:

- Account Aggregation
- Account - Group Aggregation
- Account Refresh
- Create Account
- Add/Remove entitlements
- Enable/Disable Account

Supported Managed Systems

SailPoint IdentityIQ Box.Net Connector supports the following managed systems:

- Box server

Pre-requisites

The Authentication Token is required for the Box.Net connector that needs to be managed. This has to be acquired by the Box.Net Enterprise administrator from Box.Net server to allow IdentityIQ to interact with Box.Net server. The application token and authentication token are verified with each transaction.

Administrator permissions

The administrator Should be able to create and retrieve enterprise users, enterprise user memberships, and Box groups.

Configuration parameter

Access Token: This is the authentication token.

Schema attributes

The application schema is used to map Box.Net server user or group objects to IdentityIQ account and group objects. The following Box.Net managed user and group object attributes are mapped to IdentityIQ Account and Group objects respectively.

Account attributes

The following table lists the account attributes ([Table 7—Account attributes](#)):

Table 7—Account attributes

| Attributes | Description |
|-----------------|--|
| user_id | User ID as assigned by Box.Net server. |
| name | Name of the managed user. |
| login | Email ID used to login. |
| role | User role as admin/co-admin/user. |
| space_amount | Space allocated to the user in GigaBytes. |
| shared_contacts | Whether the user can see other users in the enterprise in their contact lists as Yes/No. |
| sync_enabled | Whether the user can use synchronization. |
| job_title | The user's job title displayed on their profile page. |
| phone | The user's phone number displayed on their profile page. |
| address | User address. |
| language | User language. |

Group attributes

The following table lists the group attributes ([Table 8—Group attributes](#)):

Table 8—Group attributes

| Attributes | Description |
|--------------|---|
| group_id | Group ID as assigned by Box.Net server. |
| group_name | Group name. |
| folder_count | Group's folder count. |
| user_count | Group's user membership count. |

Box.Net group membership and group access: Whenever an entitlement is requested for a user, by default the Box.Net connector adds a user into a group with **member** as the access value. This is true for all entitlement requests.

Provisioning Policy attributes

The following table lists the provisioning policy attributes ([Table 9—Provisioning Policy attributes](#)):

Table 9—Provisioning Policy attributes

| Attributes | Description |
|---------------------|--|
| Name | Friendly name for the managed user. |
| Login Id | Email id used to login. |
| Space amount | Space to be allocated to the user in GigaBytes. |
| Shared contacts | Whether the user can see other users in the enterprise in their contact lists. |
| Role | Box.Net user role as admin/co-admin/user. |
| Language | Specify two letter code the user's language.Ex.en for English. |
| Status | Whether the user state should be active/inactive. |
| Sync enabled | Whether the user can use sync. |
| External id | Tracking code value for the SSO external ID tracking code. |
| Job title | The user's job title, displayed on their profile page. |
| Phone | The user's phone number, displayed on their profile page. |
| Address street | The street part of the user's address. |
| Address city | The city part of the user's address. |
| Address state | The state part of the user's address. |
| Address postal code | The postal code part of the user's address. |

Provisioning Policy attributes

Provisioning Policy attributes

Chapter 6: SailPoint IdentityIQ Microsoft Office 365 Connector

The following topics are discussed in this chapter:

| | |
|--|----|
| Overview | 73 |
| Supported features | 73 |
| Prerequisites | 73 |
| Administrator permissions | 74 |
| Configuration parameters | 74 |
| Schema attributes | 75 |
| Account attributes | 75 |
| Group attributes | 76 |
| Provisioning Policy attributes | 77 |
| Install and register the IQService for Windows | 78 |

Overview

This connector manages the users, groups and their attributes present on the Microsoft Office 365 Online directory store. It does not manage the attributes associated with other products in Microsoft Office 365 suite like Exchange Online, SharePoint Online, Lync Online.

The Microsoft Office 365 Connector uses Microsoft Office 365 cmdlets for Windows PowerShell to implement its functionalities in IQService which needs to be running on Windows 7 or Windows Server 2008 R2 computer.

Supported features

SailPoint IdentityIQ Microsoft Office 365 Connector provides support to the following features:

- Account Aggregation
- Account - Group Aggregation
- Account Refresh
- Create/Delete/Update Account
- Create/Delete/Update Group
- Add/Remove Entitlements
- Enable/Disable Account (Revoke/Restore)
- Password Reset
- Pass through Authentication

Prerequisites

- The IQService must be installed on Windows 7 or Windows Server 2008 R2 computer.
The restriction for the Operating System version is applied due to the Office 365 cmdlets.
For more information on installing IQService, see [“Install and register the IQService for Windows” on page 78](#).

Configuration parameters

- Office 365 cmdlets needs to be installed on a Windows 7 or Windows Server 2008 R2 computer. To install the cmdlets, perform the following steps:
 - Before installing the Office 365 cmdlets, install the Microsoft Online Services Sign-in Assistant if not already present on the system. Download and install one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 32 bit version](#)
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 64 bit version](#)
 - Download one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Module for Windows PowerShell \(32-bit version\)](#)
 - [Microsoft Online Services Module for Windows PowerShell \(64-bit version\)](#)
 - To install the cmdlets, double-click the **AdministrationConfig.msi** file. The installer adds the program to your Start menu and a shortcut to your desktop.
- Windows PowerShell and the .NET Framework 3.5.1 must be enabled.

Administrator permissions

Depending on the requested provisioning operations the user can use Office356 Connector with one of the following administrative rights:

- **User management administrator:** Can be used for most of the operations with some exceptions such as cannot delete a global administrator, create other administrators, or cannot reset passwords for billing, global, and service administrator.
- **Global administrator:** Has all administrative rights.

Configuration parameters

This section contains the information that SailPoint IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Microsoft Office 365 Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|------------------------|---|
| IQService Host | Host name of the system where IQService is installed. |
| IQService Port | Port number on which IQService is listening. Default: 5050. |
| Administrator Email | User ID or user principal name of the administrator. |
| Administrator Password | Password of the administrator. |
| Page Size | Page size. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. SailPoint IdentityIQ currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 10—Account attributes](#)):

Table 10—Account attributes

| Attributes | Description |
|---------------------------------|---|
| AlternateEmailAddresses | Alternate email address of the user (external to Microsoft Office 365). |
| BlockCredential | Whether or not the user is able to sign in. |
| City | The user's city. |
| Country | The user's country. |
| Department | The user's department. |
| DisplayName | The user's display name. |
| Fax | The user's fax number. |
| FirstName | The user's first name. |
| ImmutableID | Only returned for federated users. This is the ID that is required to be federated with Microsoft Office 365. |
| isBlackBerryUser | Returns whether or not the user has a BlackBerry device. |
| isLicensed | Whether or not the user has any licenses assigned. |
| LastDirSyncTime | The date and time of the last directory synchronization (only returned from users synced with Microsoft Office 365 through Active Directory synchronization). |
| LastName | The user's last name. |
| LicenseReconciliationNeed ed | Whether or not the user currently has a mailbox without a license. In this case, the user should be licensed within 30 days to avoid losing their mailbox. |
| Licenses | A list of the user's licenses. |
| LiveID | The user's unique login ID. |
| MobilePhone | The user's mobile phone number. |
| ObjectId | The user's unique ID. |
| Office | The user's office number. |

Table 10—Account attributes (Continued)

| Attributes | Description |
|---------------------------|--|
| OverallProvisioningStatus | Whether or not the user has been provisioned for their services. |
| PasswordNeverExpires | Whether the user's password should be forced to change every 90 days. |
| Phone Number | The user's phone number. |
| Postal Code | The user's postal code. |
| Preferred Language | The user's preferred language. |
| State | The user's state. |
| StreetAddress | The user's street address. |
| StrongPasswordRequired | Whether the user is required to set a strong password when they change their password. Strong passwords are recommended. |
| Title | The user's Job title. |
| UsageLocation | The country where the services are consumed by the user. This must be a two letter country code. |
| User ID | The user ID of the user. |
| ValidationStatus | Whether or not the user has any errors. |
| groups | It specifies all the groups to which user belongs to. |
| Role | It specifies all of the administrator roles that the specified user belongs to. |

Group attributes

The following table lists the group attributes ([Table 11—Group attributes](#)):

Table 11—Group attributes

| Attributes | Description |
|-----------------|---|
| Common Name | The group's common name. |
| Description | A description of the group. |
| Display Name | The group's display name. |
| Email Address | The group's email addresses. This is not returned for security groups. |
| Errors | A list of errors for the group. |
| Group Type | The group's type. Types can be SecurityGroup, Distributionlist or MailEnabledSecurityGroup. |
| IsSystem | Whether or not this group is a system group (created by Microsoft Office 365). These groups cannot be updated or removed. |
| LastDirSyncTime | The date and time that the group was last synched. |
| ManagedBy | The owner of the group. |
| ObjectId | The group's unique object ID. |

Table 11—Group attributes

| Attributes | Description |
|------------------|--|
| ValidationStatus | Whether or not the group has any errors. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes (Table 12—Provisioning Policy attributes):

Table 12—Provisioning Policy attributes

| Attributes | Description |
|--|--|
| Provisioning policy attributes for Account creation | |
| UserID | User ID or e-mail. This is a unique ID of the user. |
| *password* | Password for Microsoft Office365 account. |
| DisplayName | Display name of the user. |
| FirstName | First name of the user. |
| LastName | Last name of the user. |
| Department | Department of the user. |
| City | The user's city . |
| StreetAddress | Street address. |
| PostalCode | The user's postal code . |
| State | The user's state. |
| Country | Country of the user. |
| UsageLocation | The location of the user where services are consumed. Note: All user location which are present in Microsoft Office 365 are not listed here. Additional supported location can be added in the format "Country;Country code". The country code are listed at http://msdn.microsoft.com/en-us/library/ms707477%28v=vs.85%29.aspx |
| PhoneNumber | Phone number of the user. |
| MobilePhone | Mobile phone number. |
| ForceChangePassword | When true, the user will be required to change their password the next time they sign in. |
| Role | Assign administrative roles to the user. |
| AlternateEmailAddresses | Alternate email address of the user. |
| Title | User occupation. |
| Fax | Fax number of user. |
| Provisioning policy attributes for CreateGroup | |

Install and register the IQService for Windows

Table 12—Provisioning Policy attributes (Continued)

| Attributes | Description |
|--|-----------------------------------|
| DisplayName | Name of Office365 security group. |
| Provisioning policy attributes for UpdateGroup | |
| Description | A description of the group. |

Install and register the IQService for Windows

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the **IQService.zip** archive from the **IIQHome\WEB-INF\bin\win** directory of the SailPoint IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.
IQService.exe -i
It registers the service with the new registry path, **HKEY_LOCAL_MACHINE\SOFTWARE\SailPoint\IQService** with the following keys:
 - **port**: port to listen
 - **tracefile**: path to the tracefile
 - **tracelevel**: 0 (off)
3 (verbose)
4. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s
Other command line options with this service are:
 - **-d**: run in the foreground in debug mode instead of in the background using the service control manager
 - **-k**: stop the service
 - **-r**: remove the service
 - **-v**: display version information

Chapter 7: SailPoint IdentityIQ Microsoft Office 365 Exchange Online Connector

The following topics are discussed in this chapter:

| | |
|--|----|
| Overview | 79 |
| Supported features | 79 |
| Prerequisites | 79 |
| Administrator permissions | 80 |
| Configuration parameters | 80 |
| Schema attributes | 81 |
| Account attributes | 81 |
| Group attributes | 82 |
| Provisioning Policy attributes | 82 |
| Install and register the IQService for Windows | 84 |

Overview

This connector manages the user mailboxes, distribution groups, role groups and their attributes present on the Microsoft Office 365 Exchange Online store. It does not manage the attributes associated with other products in Microsoft Office 365 suite like Microsoft Office 365 Online directory store, SharePoint Online, Lync Online.

The Microsoft Office 365 Exchange Online Connector uses Microsoft Office 365 Exchange cmdlets for Windows PowerShell to implement its functionalities in IQService which needs to be running on Windows 7 or Windows Server 2008 R2 computer.

Supported features

SailPoint IdentityIQ Microsoft Office 365 Exchange Online Connector provides support for the following features:

- Account Aggregation
- Account Group Aggregation
- Account Refresh
- Create/Delete/Update Account
- Create/Delete/Update Group
- Add/Remove entitlements

Prerequisites

- The IQService must be installed on Windows 7 or Windows Server 2008 R2 computer.
The restriction for the Operating System version is applied due to the Office 365 cmdlets.

For more information on installing IQService, see [“Install and register the IQService for Windows” on page 84](#).

Configuration parameters

- Office 365 cmdlets needs to be installed on a Windows 7 or Windows Server 2008 R2 computer. To install the cmdlets, perform the following steps:
 - Before installing the Office 365 cmdlets, install the Microsoft Online Services Sign-in Assistant if not already present on the system. Download and install one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 32 bit version](#)
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 64 bit version](#)
 - Download one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Module for Windows PowerShell \(32-bit version\)](#)
 - [Microsoft Online Services Module for Windows PowerShell \(64-bit version\)](#)
 - To install the cmdlets, double-click the **AdministrationConfig.msi** file. The installer adds the program to your Start menu and a shortcut to your desktop.
- Windows PowerShell and the .NET Framework 3.5.1 must be enabled.

Administrator permissions

Depending on the requested provisioning operations the user can use Office356 Connector with one of the following administrative rights:

- **User management administrator:** Can be used for most of the operations with some exceptions such as cannot delete a global administrator, create other administrators, or cannot reset passwords for billing, global, and service administrator.
- **Global administrator:** Has all administrative rights.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Microsoft Office 365 Exchange Online Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|------------------------|--|
| IQService Host | Host name of the system where IQService is installed. |
| IQService Port | Port number on which IQService is listening. Default: 5050. |
| Administrator Email | User ID or user principal name of the administrator. |
| Administrator Password | Password of the administrator. |
| Page Size | The number of objects to fetch in a single page when iterating over large data sets. Default: 500. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 13—Account attributes](#)):

Table 13—Account attributes

| Attributes | Description |
|-------------------------------|---|
| Alias | Alias (mail nickname) of the user. |
| AntispamBypassEnabled | Whether or not to skip anti-spam processing on this mailbox. |
| Database | The database that contains the mailbox object. |
| DeliverToMailboxAndForward | Whether messages sent to this mailbox are forwarded to another address. |
| DisplayName | The user's display name. |
| DistinguishedName | Distinguished name. |
| EmailAddresses | Email addresses. |
| EmailAddressPolicyEnabled | Whether or not the e-mail address policy for this mailbox is enabled. |
| ExchangeGuid | Exchange Guid. |
| ExchangeUserAccountControl | Exchange user account flags. |
| ExchangeVersion | Exchange version. |
| HiddenFromAddressListsEnabled | Whether or not this mailbox is hidden from other address lists. |
| Identity | The Identity parameter specifies the mailbox. |
| IsMailboxEnabled | Whether or not mailbox is enabled. |
| LegacyExchangeDN | Legacy exchange distinguished name. |
| Name | Attribute name for the mailbox. |
| OrganizationalUnit | Organizational unit. |
| SamAccountName | The SamAccountName parameter specifies the user name for earlier operating systems. |
| SimpleDisplayName | Simple display name. |
| UMEnabled | Whether or not unified messaging is enabled for the account. |
| UserID | User ID or e-mail of the user. |

Table 13—Account attributes (Continued)

| Attributes | Description |
|---------------|---|
| groups | It specifies all the groups to which user belongs to. |
| RecipientType | Recipient type. |
| AccountSkuld | Licensing plans that are available. |

Group attributes

The following table lists the group attributes ([Table 14—Group attributes](#)):

Table 14—Group attributes

| Attributes | Description |
|-------------------------|---|
| DisplayName | The group's display name. |
| DistinguishedName | Distinguished name of a group. |
| EmailAddresses | The group's email addresses. This is not returned for security groups. |
| Group Type | The group's type. Types can be SecurityGroup, Distributionlist or MailEnabledSecurityGroup. |
| Alias | Alias. |
| GroupName | Name of the group. |
| members | Users assigned to a group. |
| MemberDepartRestriction | If anyone can leave this group without being approved by the group owners.(Applicable for Distribution Group.) |
| MemberJoinRestriction | If anyone can join this group without being approved by the group owners. (Applicable for Distribution Group.). |
| PrimarySmtpAddress | The Primary SMTP email address (Applicable for Distribution Group). |
| Description | Description of Role Group. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes ([Table 15—Provisioning Policy attributes](#)):

Table 15—Provisioning Policy attributes

| Attributes | Description |
|--|--|
| Provisioning policy attributes for Account creation | |
| UserID | First name or e-mail. This is a unique ID of the user. |
| DisplayName | Display name of the User. |

Table 15—Provisioning Policy attributes (Continued)

| Attributes | Description |
|---|---|
| AccountSkuld | Licensing plans that are available for exchange (AccountSkuld). These plans can be retrieved by executing the powershell cmdlet Get-MsolAccountSku and it will be displayed in domain:plan format under the AccountSkuld column. Note: Before provisioning, it is mandatory to replace/modify the default values of 'AccountSkuld' in provisioning policy with the values retrieved as mentioned above. |
| ProhibitSendReceiveQuota | Mailbox size at which the user associated with this mailbox can no longer send or receive messages. |
| ProhibitSendQuota | It specifies the mailbox size at which the user associated with this mailbox can no longer send messages. |
| MailTip | The MailTip parameter specifies the message displayed to senders when they start drafting an e-mail message to this recipient. The MailTip parameter message must be less than or equal to 250 characters. |
| IssueWarningQuota | The IssueWarningQuota parameter specifies the mailbox size at which a warning message is sent to the user. You must specify either an integer or unlimited. |
| Provisioning policy attributes for CreateGroup | |
| DisplayName | Name of Distribution or Role group to be created. Note: While creating a new Role Group for the first time on the Exchange Online, an error message is displayed that asks you to run the Enable-OrganizationCustomization cmdlet. To resolve this, issue create a new dummy Role Group on Exchange Online or run the Enable-OrganizationCustomization Cmdlet from Powershell. (This operation is required only once). This is a known behaviour as per Microsoft documentation http://help.outlook.com/beta/hh299030.aspx . |
| GroupType | Indicates group type (Distributed Group or Role Group). |
| Provisioning policy attributes for UpdateGroup | |
| GroupType | Indicates group type (Read only attribute). |
| Alias | The Alias parameter specifies the alias of the distribution group. The Alias parameter is used to generate the primary SMTP e-mail address of the object. The value of the Alias parameter cannot contain spaces. |
| MemberDepartRestriction | If anyone can leave this group without being approved by the group owners (Applicable for Distribution Group). Allowed values are: 1. Open 2. Closed |

Install and register the IQService for Windows

Table 15—Provisioning Policy attributes (Continued)

| Attributes | Description |
|-----------------------|--|
| MemberJoinRestriction | If anyone can join this group without being approved by the group owners. (Applicable for Distribution Group.). Allowed values are 1. Open 2. Closed |
| PrimarySmtpAddress | The Primary SMTP email address (Applicable for Distribution Group). |
| Description | Description of Role Group. |

Install and register the IQService for Windows

1. Create a directory in which you want to download the service. For example, `c:\iqservice`.
2. Extract the **IQService.zip** archive from the `IIQHome\WEB-INF\bin\win` directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.
IQService.exe -i
It registers the service with the new registry path, `HKEY_LOCAL_MACHINE\SOFTWARE\SailPoint\IQService` with the following keys:
 - **port**: port to listen
 - **tracefile**: path to the tracefile
 - **tracelevel**: 0 (off)
3 (verbose)
4. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s
Other command line options with this service are:
 - **-d**: run in the foreground in debug mode instead of in the background using the service control manager
 - **-k**: stop the service
 - **-r**: remove the service
 - **-v**: display version information

Chapter 8: SailPoint IdentityIQ ServiceNow Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|----|
| Overview..... | 85 |
| Supported features | 85 |
| Pre-requisites | 85 |
| Administrator permissions | 86 |
| Configuration parameters..... | 86 |
| Schema attributes | 87 |
| Account attributes | 88 |
| Group attributes..... | 89 |
| Provisioning Policy attributes | 90 |
| Additional information | 91 |
| Configuration settings | 91 |
| Troubleshooting..... | 92 |

Overview

The ServiceNow connector manages ServiceNow accounts and groups. It supports read and write for ServiceNow accounts.

Supported features

SailPoint IdentityIQ ServiceNow Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Create/Update/Delete User
- Enable/Disable/Unlock User
- Change password
- Add/Remove Entitlement(Account-group, roles)
- Create/Update/Delete Group

Pre-requisites

ServiceNow should be up and running.

Administrator permissions

User should have the following privileges:

- sys_user: Full Permission
- sys_user_group: Read Only
- sys_user_role: Read Only
- sys_user_grmember: Full Permission
- sys_user_has_role: Full Permission
- cmn_department: Read Only
- cmn_location: Read only
- sys_choice: Read only
- cmn_cost_center: Read only
- sys_group_has_role: Full permission

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The ServiceNow Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|----------------|---|
| serviceNow URL | The URL for ServiceNow. |
| UserName | User name of the administrator having the following privileges: -> sys_user: Full Permission -> sys_user_group: Read Only -> sys_user_role: Read Only -> sys_user_grmember: Full Permission -> sys_user_has_role: Full Permission -> cmn_department: Read Only -> cmn_location: Read only -> sys_choice: Read only -> cmn_cost_center: Read only -> sys_group_has_role: Full permission |
| Password | Password of the administrator user. |

| Parameters | Description |
|---------------------|---|
| Authentication Type | <ul style="list-style-type: none"> ■ Basic/WS-Security: Select WS-Security if WS-Security is enabled on ServiceNow side. In case of Basic, username/password will be used for authentication. In case of WS-Security, certificate from keystore will be used. <p>Note: The WS-Security method uses the 'Crypto Caching' feature, when WS-Security method has been selected for ServiceNow Connector. If Crypto Caching is enabled, the crypto objects are read from a cache instead of constructing them by reading the keystore files. Enabling caching of crypto objects improves the performance of security processing.</p> <p>This means that once you have one successful test connection performed, the crypto objects are cached (crypto objects are nothing but the parameters specified in application console for certificate based authentication). Now, if you have modified the crypto parameters from application console with incorrect values, then:</p> <ul style="list-style-type: none"> - all the operations except test connection are successful as the values are taken from the cache. - Only the test connection will fail in this scenario as test connection operation does not refer to cache but try to validate the parameters specified by user. <ul style="list-style-type: none"> ■ Basic and WS Secured: Select this option when ServiceNow is configured to support Basic and WS Secured Authentications together. ■ Username token and WS Secured: Select this option when ServiceNow is configured to support Username token and WS Secured Authentication. <p>Note: Each time you change the Authentication Type for ServiceNow connector ensure that you perform Test Connection operation.</p> |
| Keystore Path | Full path of the generated keystore. |
| Keystore Password | Password of the keystore. |
| Keystore Type | Type of the certificate. |
| Certificate Alias | Alias of the certificate. |
| Alias Password | Password of the alias. |

Note: User must configure their ServiceNow with **Basic/WS-Security**, **Basic and WS Secured**, or **Username token and WS Secured** authentication. When the authentication type is not selected security threats are imposed.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 16—Account attributes](#)):

Table 16—Account attributes

| Attributes | Description |
|----------------------|--|
| first_name | First name of the user. |
| last_name | Last name of the user. |
| email | Email ID of the user. |
| user_name | Name of the user. |
| Department | The user's department name. |
| title | Title(designation) of the user. |
| sys_id | Unique ID generated by system for user. |
| Phone | Phone number of user. |
| calendar_integration | Determines whether change requests assigned to that user are sent to their Outlook calendar. |
| sys_class_name | Class name of the user. |
| company | The user's company. |
| cost_center | Cost centre of the user. |
| sys_created_on | Date this user is created in ServiceNow. |
| sys_created_by | Administrator who created the user in ServiceNow. |
| groups | List of groups the user is part of. |
| roles | List of roles the user is part of. |
| active | Determines whether the user account has been staged for use. |
| building | The building of the user. |
| city | The city of the user. |
| country | The country of the user. |
| Location | The location of the user. |
| manager | The manger of the user. |
| middle_name | Middle name of the user. |
| name | Name of the user. |
| password_needs_reset | Determines should the user be prompted to change password at next login. |
| user_id | User ID for the user. |
| default_perspective | Default perspective for the user. |
| sys_domain | Domain of the user. |
| employee_number | Employee number of the user. |

Table 16—Account attributes (Continued)

| Attributes | Description |
|--------------------|---|
| failed_attempts | Number of login failed attempts. |
| gender | Gender of the user. |
| home_phone | Home phone number of the user. |
| ldap_server | LDAP server the user has an account. Identifies which LDAP server authenticates the user when there are multiple LDAP servers. |
| preferred_language | Language spoken for the user. |
| last_login | Last login date of the user. |
| last_login_time | Time of the last login time for the user. |
| locked_out | Determines if user account is locked. |
| mobile_phone | Mobile number of the user. |
| notification | Determines if the user should be notified for any changes made on his account. |
| schedule | Schedule of the user. |
| state | The state for the user. |
| source | Identifies whether or not LDAP is used to validate a user. If the Source field starts with ldap , then the user is validated via LDAP. If the Source field does not start with ldap , then the password on the user record is used to validate the user upon login. |
| street | The street for the user. |
| time_format | Time format chosen for user to display time fields. |
| time_zone | The timezone for the user. |
| sys_updated_on | Last updated time for the user. |
| sys_updated_by | The last update for the user occurred from. |
| sys_mod_count | Number of updates for the user. |
| vip | Determines if the user is treated as VIP. |
| Zip | Zip for the user. |

Group attributes

The following table lists the group attributes (Table 17—Group attributes):

Table 17—Group attributes

| Attributes | Description |
|----------------|--|
| active | Determines whether the user account has been staged for use. |
| cost_center | Cost centre of the user group. |
| sys_created_on | Date the user group is created in ServiceNow. |

Provisioning Policy attributes

Table 17—Group attributes

| Attributes | Description |
|------------------|---|
| sys_created_by | Administrator who created the user in ServiceNow. |
| default_assignee | Defaults assignee for the user group. |
| description | Description of the user group |
| exclude_manager | Determines if the manager should be excluded for the use group. |
| name | Name of the user group. |
| parent | Parent group of this user group. |
| roles | Roles the user group is having. |
| source | Source of the user group. |
| sys_id | Unique ID generated by system for user group. |
| type | Type of the user group |
| sys_updated_on | Last updated time for the user group. |
| sys_updated_by | The last update for the user group occurred from. |
| sys_mod_count | Number of updates for the user group. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes for create ([Table 18—Provisioning Policy attributes](#)):

Table 18—Provisioning Policy attributes

| Attributes | Description |
|----------------------|--|
| User ID | User ID for the user. |
| First Name | First name of user. |
| Last Name | Last name of the user. |
| Department | The user's department name. |
| Title | Title(designation) of the user |
| Password | Password for the user. |
| Password need reset | Determines should the user be prompted to change password at next login. |
| Locked Out | Determines if user account is locked. |
| Active | Determines whether the user account has been staged for use. |
| Notifications | Determines if the user should be notified for any changes made on his account. |
| Calender integration | Determines whether change requests assigned to that user are sent to their Outlook calendar. |
| Time Zone | The time zone for the user. |

Table 18—Provisioning Policy attributes (Continued)

| Attributes | Description |
|----------------|-----------------------------|
| Email | Email of the user. |
| Mobile Phone | Mobile number of the user. |
| Business Phone | Official phone of the user. |

Additional information

This section describes the additional information related to the ServiceNow Connector.

Configuration settings

Timeout for connection between ServiceNow and IdentityIQ can be set in application template using the debug page from IdentityIQ. Default value is 10 minutes.

For example, entry: `<entry key="socketTimeout" value="4"/>`

The above example will set the connection timeout to 4 minutes.

Note: The value mentioned for **socketTimeout** should be less than or equal to the request timeout set on ServiceNow represented by the **glide.soap.request_processing_timeout** parameter in **sys_properties.list** file.

If aggregation fails due to connection timeout, ServiceNow will internally retry the aggregation. Number of retry attempts can be set in application template using the debug page from IdentityIQ. Default value is 3.

For example, entry, `<entry key=" aggregation_retry_attempts" value="4"/>`

The above example will set the max retry attempts to 4.

Supporting the role attribute

Entitlements in ServiceNow has a field called **Roles**. This field indicates the Roles connected to that Role. Supporting this Role field on ServiceNow Connector can cause performance degradation for Account-Group Aggregation task. To support the **Roles** field, set the **ManageRolesforGroup** parameter as follows on the Application Config for ServiceNow:

entry: `<entry key="ManageRolesforGroup" value="true"/>`

Setting the **ManageRolesforGroup** parameter as above, supports Role attribute for Group. Default value for this is false which means **Roles** field have no value after aggregation and any type of modification is not supported on the ServiceNow System.

Viewing the users connected to an account-group

Add **groups** attribute in **account mappings** for a ServiceNow application in order to view the users connected to an account-group. Perform the following procedure:

1. Navigate to **System setup => Account Mappings**.
2. Add an account attribute.
3. Add source mapping.

4. Select an application.
5. Select **groups** attribute.
6. Add and save the **groups** attribute.

Troubleshooting

1 - Whenever a new field is created to any of the following tables, the new fields are ignored by ServiceNow Connector

- sys_user
- sys_user_grmember
- sys_user_has_role
- sys_user_role
- sys_user_group
- cmn_department
- cmn_location
- sys_choice
- cmn_cost_center: Read only
- sys_group_has_role: Full permission

ServiceNow Connector is unaware of the new created fields and hence are ignored. In order to get the values of the newly created fields, perform the steps mentioned in the “Generate Stub classes for ServiceNow connector” section on [page 92](#) below.

Generate Stub classes for ServiceNow connector

If any of the following tables are customized on ServiceNow, the stub classes for respective **wsdl** must be generated and replaced in **iiq.jar** file.

- sys_user
- sys_user_grmember
- sys_user_has_role
- sys_user_role
- sys_user_group
- cmn_department
- cmn_location
- sys_choice
- cmn_cost_center: Read only
- sys_group_has_role: Full permission

Perform the following procedure to generate stub classes for a wsdl:

1. Get the **sys_user wsdl** file from ServiceNow using following url and save it to local machine.
https://servicenowinstance.service-now.com/sys_user.do?WSDL
2. Download apache Axis2 version 1.6.1
<http://axis.apache.org/axis2/java/core/download.cgi>
3. Extract downloaded **axis2** file and set the **AXIS2_HOME** environment variable pointing to this directory.
4. Set **JAVA_HOME** pointing to the jdk directory.
5. Navigate to **%AXIS2_HOME%\bin** and run the following command:
wsdl2java.bat -uri <wsdl file path> -Eosv -p openconnector.connector.servicenow -o <path>

Following files are generated:

- ServiceNow_sys_userStub.java
- ServiceNow_sys_userCallbackHandler.java in `<path>/src/openconnector/connector/servicenow`

6. To compile these classes, add the following jar files from `%AXIS2_HOME%\lib` into the classpath along with the current directory.

set

```
classpath=.;%AXIS2_HOME%\lib\axiom-api-1.2.12.jar;%AXIS2_HOME%\lib\axiom-impl-1.2.12.jar;%AXIS2_HOME%\lib\axis2-adb-1.6.1.jar;%AXIS2_HOME%\lib\axis2-kernel-1.6.1.jar;%AXIS2_HOME%\lib\axis2-transport-http-1.6.1.jar;%AXIS2_HOME%\lib\axis2-transport-local-1.6.1.jar;%AXIS2_HOME%\lib\geronimo-stax-api_1.0_spec-1.0.1.jar;%AXIS2_HOME%\lib\neethi-3.0.1.jar;%AXIS2_HOME%\lib\wsdl4j-1.6.2.jar;%AXIS2_HOME%\lib\wstx-asl-3.2.9.jar;%AXIS2_HOME%\lib\XmlSchema-1.4.7.jar
```

7. Compile the classes using the following command:

```
%JAVA_HOME%\bin>javac <path>\src\openconnector\connector\servicenow\*.java
```

8. Copy the newly generated stub classes from `\src\openconnector\connector\servicenow` into `/identityIQ/WEB-INF/classes/openconnector/connector/servicenow`

For example, copy `<path>\src\openconnector\connector\servicenow*.class`
`\identityIQ\WEB-INF\classes\openconnector\connector\servicenow\`

2 - When you upgrade IdentityIQ from version 5.5 patch 2 a or above to version 6.0, the new fields are ignored by ServiceNow Connector

ServiceNow Connector is unaware of the new created fields and hence are ignored when you upgrade IdentityIQ version 5.5 patch 2 or above to version 6.0.

Workaround: Perform the steps mentioned in the “Generate Stub classes for ServiceNow connector” section on [page 92](#) even if the Generate Stub classes steps were performed earlier on the previous installed patch.

3 - Account entity has a ‘groups’ field which displays all the groups the user is a part of.

The Account entity has a groups field which displays all the groups the user is a part of ; the Account aggregation only displays the sys id of those groups that is, after aggregation the account details display some alpha numeric strings in the **groups** field.

Workaround: In order to get the group name, the account-group aggregation must be executed, which will replace the sys id from corresponding group name in the **groups** field.

4 - When IdentityIQ is configured with Weblogic version 10.3 or below, jrockit 1.5/1.6; IdentityIQ initialization fails.

When IdentityIQ is configured with Weblogic version 10.3 or below, jrockit 1.5/1.6, IdentityIQ failed with the following exception:

```
.....
.....
at weblogic.t3.srvr.SubsystemRequest.run(SubsystemRequest.java:64)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:201)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:173)
java.lang.ClassCastException: com.ctc.wstx.stax.WstxInputFactory
at
javax.xml.stream.XMLInputFactory.newInstance(XMLInputFactory.java:137)
.....
.....
```

Troubleshooting

Following is an identified defect in Oracle Weblogic:

Bug 8176755: [CR385303] WLS 10.3 - CLASSCASTEXCEPTION WHEN DEPLOYING APPLICATION CONTAINING STAX IMPLEMENTATION.

Doc Id for the same in Oracle KB is ID 1134184.1.

Workaround: Following patches are available for 8176755 bug:

Table 19—Patch repository information

| WLS version | Patch ID | Passcode |
|-------------|----------|----------|
| 10.3.0 | IQXV | HJZPM5SC |
| 10.3.1 | BR4K | 42JMWTA |
| 10.3.2 | JKDI | Z4372ITJ |

5 - When WS-Security method is selected in ServiceNow instance and Basic authentication is selected in ServiceNow connector an error message is displayed.

When WS-Security method is selected in ServiceNow instance and Basic authentication is selected in ServiceNow connector, the following error message is displayed:

```
openconnector.ConnectorException:org.apache.axis2.AxisFault:soap message must contain a document type declaration.
```

This error generally occurs when you have selected certificate based authentication on ServiceNow instance and in ServiceNow connector it is selected as Basic Authentication.

Workaround: Verify the authentication method selected on ServiceNow connector and match it to the authentication method selected by the ServiceNow instance.

6 - For certificate based authentication the IdentityIQ Server and ServiceNow instance must have correct time set.

For certificate based authentication, ensure that the IdentityIQ Server and ServiceNow instance have the correct date, time and timezone set.

7 - When the Test Connection fails error messages are displayed in IdentityIQ on JBoss and WebSphere

- **For WebSphere**

The following error message is displayed in IdentityIQ:

```
Unable to engage module : rampart
```

The following error message is displayed in the log file of WebSphere:

```
ERROR WebContainer : apache.axis2.deployment.ModuleDeployer:113 - The rampart-1.6.1.mar module, which is not valid, caused Could not initialize class org.apache.axis2.deployment.util.TempFileManager
```

Resolution: If the above error message is displayed in the log file of WebSphere, set the temporary directory in **Generic JVM arguments** of **Java Virtual Machine** by setting the following variable:

-Djava.io.tmpdir=<FullPathOfTempDir>

Note: Ensure that the UNIX user where WebSphere is installed should be the owner of the temporary directory.

- **For JBoss**

The following error message is displayed in IdentityIQ:

Unable to engage module : rampart

Resolution: If the above error message is displayed in IdentityIQ, copy the following files from iiq\WEB-INF\lib\ to deploy directory of JBoss in order to work with certificate based authentication on JBoss (Windows only):

rahas-1.6.1.mar

rampart-1.6.1.mar

Chapter 9: SailPoint IdentityIQ GoToMeeting Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|----|
| Overview | 97 |
| Supported features | 97 |
| Pre-requisites | 97 |
| Administrator permissions | 97 |
| Configuration parameter | 97 |
| Schema attributes | 98 |
| Account attributes | 98 |
| Group attributes | 98 |
| Provisioning Policy attributes | 99 |

Overview

SailPoint GoToMeeting connector manages GoToMeeting organizers. It supports read and write to GoToMeeting to create, retrieve, update, delete users, and retrieve groups.

Supported features

GoToMeeting supports the following features:

- Account Aggregation
- Account Group Aggregation
- Create Account
- Read Account (except invited accounts)
- Enable/Disable Account (except invited accounts)
- Delete Account (except invited accounts)

Pre-requisites

The user will be walked through the OAuth2 flow to generate the access token using the AccessIQ and then pass it down to the IdentityIQ GoToMeeting connector. The connector will use this Access Token to make calls to any GoToMeeting REST API.

Administrator permissions

Role of the user must be an Administrator.

Configuration parameter

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Schema attributes

Access Token: A valid Access Token for the user is required which enables your application to access the user's information and take actions on their behalf. The application and user are verified with each API call by passing an access token along with each request.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 20—Account attributes](#)):

Table 20—Account attributes

| Attributes | Description |
|-------------------------|---|
| OrganizerKey | A unique key associated with each organizer. |
| FirstName | First Name of the organizer. |
| LastName | Last Name of the organizer. |
| Email | Email id of the organizer |
| Status | The status of the organizer (Active, Invited or Suspended). |
| GroupKey | A unique key associated with a group of which the organizer is a part of. |
| Groups | The entitlements of the organizer. |
| MaximumAttendeesAllowed | The maximum number of attendees allowed for the organizer. |

Group attributes

The following table lists the group attributes ([Table 21—Group attributes](#)):

Table 21—Group attributes

| Attributes | Description |
|--------------------|---|
| GroupName | The name of the group. |
| GroupKey | A unique key associated with the group. |
| ParentKey | The Parent Key of the group. |
| GroupStatus | The status of the group. |
| NumberOfOrganizers | The number of organizers in a group. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes for create ([Table 22—Provisioning Policy attributes](#)):

Table 22—Provisioning Policy attributes

| Attributes | Description |
|----------------|---|
| OrganizerEmail | A valid email id is required to whom a GoToMeeting invite should be sent. |

Provisioning Policy attributes

Chapter 10: SailPoint IdentityIQ Microsoft SharePoint Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 101 |
| Supported features | 101 |
| Supported Managed system | 102 |
| Pre-requisites | 102 |
| Administrator permissions | 102 |
| Configuration parameters | 102 |
| Schema attributes | 103 |
| Account attributes | 103 |
| Group attributes | 103 |
| Provisioning Policy attributes | 104 |
| Install and register the IQService for Windows | 105 |
| Install and register the IQService | 105 |
| Additional information | 105 |
| Unstructured Target Collector | 105 |
| Troubleshooting | 107 |

Overview

Microsoft SharePoint provides tools which provides users the ability to set up websites to share information with others, manage documents from start to finish, and publish reports to help everyone make better decisions.

IdentityIQ offers the ability to aggregate existing SharePoint users from any SharePoint site or collection and display what access rights those SharePoint users have to SharePoint groups, sites, lists, folders, and files.

The Microsoft SharePoint connector is designed to manage SharePoint users and groups from SharePoint 2007, 2010 (Classic mode or Windows Claim based authentication) and 2013 (Windows claim based authentication) environments using the Microsoft SharePoint server APIs which ship as part of the SharePoint software. Presently, the domain groups which are considered as users in SharePoint are not supported through the connector.

Supported features

The SharePoint connector provides the ability to provision users, groups, and entitlements from IdentityIQ. The connector supports the following functions:

- Users and groups aggregation
- Create/Delete/Update User
- Create/Delete/Update Group
- Add/Remove entitlements
- Read/Revoke Unstructured Target permissions for Sites, Lists, List Items, Folders, and Files.
- Target Aggregation

For more information see, [“Unstructured Target Collector”](#) section.

Note: Before you can use any of the features of the connector, the IQService must be installed and registered on SharePoint Server. For more information about installing IQService see, [“Install and register the IQService for Windows”](#) on page 105.

Supported Managed system

Microsoft SharePoint connector supports following Microsoft SharePoint servers:

- Microsoft SharePoint server 2007
- Microsoft SharePoint server 2010 (Classic mode or Windows Claim based authentication)
- Microsoft SharePoint server 2013 (Windows claim based authentication)

Pre-requisites

IQService must be installed and registered on Windows Host computer where SharePoint Server is configured.

Administrator permissions

Microsoft SharePoint connector is designed to manage SharePoint users and groups for this you need to provide credentials of user which is having Site Collection Administrative privileges.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SharePoint Connector uses the connection attributes listed in the following table:

| Parameters | Description |
|--------------------------------|---|
| SharePoint site collection URL | URL to the site collection from which you want to aggregate. |
| User Name | User with Site Collection Administrator permission. The user name format should be as present in SharePoint. To get user name format in SharePoint, navigate to site page, click user name in the upper-left corner, and then click My Settings . Check the Account field. For Windows Claim based authentication, the user name should be in encoding format. For example, i:0#.w contoso\chris |
| Password | The password associated with the User Name. |
| IQService Host | Host name where IQService is installed. The IQService needs to be installed on the SharePoint server. |
| IQService Port | Port number used by the IQService (Default : 5050). |
| SharePoint Server Version | Version of the SharePoint Server (Default: 2007). |
| Page Size | The number of objects to get per page, when iterating over large numbers of objects. (Default : 500). |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes ([Table 23—Account attributes](#)):

Table 23—Account attributes

| Attributes | Description |
|------------------|--|
| AccountName | Login name of the user. |
| DisplayName | Display name of the user. |
| UserName | sAMAccountName of the user. |
| Title | Title of the user. |
| FirstName | First name of the user. |
| LastName | Last Name of the user. |
| PreferredName | Preferred name of the user. |
| WorkPhone | Office phone number of the user. |
| Fax | Fax number of the user. |
| Email | E-mail address of the user. |
| WorkEmail | Office email address of the user. |
| Office | Office location of the user. |
| Manager | Manager of the user. |
| Groups | Collection of groups of which the user is a member. |
| OwnedGroups | Groups that the user owns. |
| IsSiteAuditor | Specifies whether the user is a site collection auditor. |
| IsSiteAdmin | Specifies whether the user is a site collection administrator. |
| ID | Member ID for the user. |
| SID | Unique security ID for the network account of the user. |
| UserProfile_GUID | GUID of the User Profile. |

Group attributes

The following table lists the group attributes ([Table 24—Group attributes](#)):

Table 24—Group attributes

| Attributes | Description |
|------------|--------------------|
| GroupName | Name of the group. |

Table 24—Group attributes

| Attributes | Description |
|--------------------------------|--|
| DisplayName | Name of the group. |
| Description | Description for the group. |
| ID | Identifier (ID) for the group. |
| Owner | Name of the group owner. |
| RequestToJoinLeaveEmailSetting | Membership requests to this e-mail address. |
| AutoAcceptRequestToJoinLeave | Whether membership requests are automatically accepted. |
| OnlyAllowMembersViewMembership | Who can view the membership of the group. |
| AllowRequestToJoinLeave | Whether to allow users to request for membership of the group. |
| AllowMembersEditMembership | Who can edit the membership. |

Provisioning Policy attributes

This following table lists the provisioning policy attributes ([Table 25—Provisioning Policy attributes](#)):

Table 25—Provisioning Policy attributes

| Attributes | Description |
|--|---|
| Provisioning policy attributes for Account creation | |
| AccountName | Login name of the user. The user should exist in the configured SharePoint user store (for example, Active Directory). For Windows Claim based authentication, the user name should be in encoding format. For example, i:0#.w contoso\chris |
| Provisioning policy attributes for Group creation | |
| GroupName | Name of the group. |
| Permission | Group permission. |
| DefaultUser | Default user of group. |
| Provisioning policy attributes for Group update | |
| Description | Group description. |
| Owner | Group owner. |
| RequestToJoinLeaveEmailSetting | Membership requests to this e-mail address. |
| OnlyAllowMembersViewMembership | Who can view the membership of the group. |
| AutoAcceptRequestToJoinLeave | Whether membership requests are automatically accepted. |

Table 25—Provisioning Policy attributes (Continued)

| Attributes | Description |
|----------------------------|--|
| AllowRequestToJoinLeave | Whether to allow users to request for membership of the group. |
| AllowMembersEditMembership | Who can edit the membership. |

Install and register the IQService for Windows

You must install and register an IQService before you can provision to SharePoint Connector. The IQService is a native Windows service that enables IdentityIQ to participate in a Windows environment and access information only available through Windows APIs.

IQService must be installed and registered on Windows Host computer where SharePoint Server is configured.

Install and register the IQService

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the **IQService.zip** archive from the **IIQHOME\WEB-INF\bin\win** directory of the IdentityIQ installation into the created directory.
3. To enable provisioning SharePoint 2013, rename **app.config** file located at IQService home to **IQService.exe.config** file.
4. Run the following command to install a Windows service named IQService.
IQService.exe -i
5. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information

Additional information

This section describes the additional information related to the Microsoft SharePoint Connector.

Unstructured Target Collector

SharePoint uses a data structure which requires the configuration of the **Unstructured Targets** tab to collect targeted data and correlates it with AccountName for Accounts and DisplayName for groups.

Additional information

For SharePoint target permission, the **Unstructured Targets** functionality will be enabled if SHAREPOINT_TARGET feature string is present in the application.

Multiple target sources can be specified and configured for applications which supports unstructured targets. This will be useful in the applications where the target permissions can be fetched from multiple target sources. For example, as stated in overview section, SharePoint connector does not manage domain groups. For assigning the SharePoint target permission to domain groups, SharePoint target collector can be configured for Active Directory application along with Windows file share target collector.

SharePoint Target Collector supports aggregation for Sites, Lists, List Items, Folders and Files. The objects can be filtered based on various filters configured on the Unstructured Targets Tab.

| Attribute | Description | Possible values |
|---------------------------|---|--|
| IQService Host | Host name/IP Address of the computer where IQService is installed. The IQService needs to be installed on the SharePoint server. | |
| IQService Port | Port number used by the IQService. | Default: 5050 |
| SharePoint Server Version | Version of the SharePoint Server | Default: 2007 |
| Site Collection URL | URL of Site or Site Collection for target aggregation. | URL. Cannot be blank |
| UserName | User with Site Collection Administrator permission. | The user name format should be as present in SharePoint. For Windows Claim based authentication, the user name should be in encoding format. For example, i:0#.w contoso\chris |
| Password | Password for UserName | |
| Target Types Filter | As mentioned above, the Target Collector supports aggregating Sites, Lists, List Items and Files. Using this filter, any of these target types can be selectively aggregated. | Any combination of following separated by comma: Sites,Files,Lists,ListItems,Folders,Files List Item specific filtration, for example, Document, Picture, Wiki Page and so on. If not specified, all target types would be aggregated. Default – Not specified |
| Site Filter Type | This is used in combination to the Site Filter. This tells whether the Site Filter define the inclusion filter or exclusion filter. | Include/Exclude Default: Include |

| Attribute | Description | Possible values |
|-------------|---|--|
| Site Filter | Targets with path containing Words / phrases mentioned here can be selectively included or excluded depending on the Site Filter Type | Words/phrases separated by comma. If not specified all the targets would be aggregated. Default: Not specified |

Note: By default, the connector does not support revoking limited Access permission. Limited Access permission is a special permission in SharePoint and cannot be removed explicitly. Only opting for 'Remove User Permissions' removes the Limited Access permissions.

Connector provides configuration to support similar functionality. Setting 'RevokeLimitedAccess' to "true" in the application attributes enables the connector to 'Remove User Permissions' when revocation for Limited Access permission is requested. This will remove all permissions assigned to the user for the resource and all resources below it. By default 'RevokeLimitedAccess' is set to false.

Troubleshooting

1 - User profile attributes are not aggregated during account aggregation.

Due to insufficient rights on user profile application, user profile attributes are not aggregated during account aggregation. IQService displays the following error message:

```
UserProfile can't retrieve
```

Resolution: Provide full access control permission on user profile application, by performing the following:

1. Navigate to **Central Admin ==> Manage service Applications.**
2. Select **User Profile Service Application.**
3. On ribbon select **Permissions** and add your account and provide **Full Control rights** .

Chapter 11: SailPoint IdentityIQ Amazon Web Services Identity and Access Management Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 109 |
| Supported features | 110 |
| Pre-requisites | 110 |
| Administrator permissions | 111 |
| Schema attributes | 111 |
| Account schema | 111 |
| Group schema | 112 |
| Provisioning Policy attributes | 112 |
| Account | 112 |
| Account-Group | 113 |
| Additional information | 113 |
| Amazon Web Services Identity and Access Management API's | 113 |
| Troubleshooting | 115 |

Overview

Amazon Web Services (AWS) Identity and Access Management (IAM) helps you securely control access to Amazon Web Services and your account resources. With IAM, you can create multiple IAM users under your AWS account or enable temporary access through identity federation with your corporate directory. In some cases, you can also enable access to resources across AWS accounts. IAM offers greater security, flexibility, and control when using AWS.

Without IAM, however, you must either create multiple AWS accounts—each with its own billing and subscriptions to AWS products—or share the security credentials of a single AWS account. In addition, without IAM, you cannot control the tasks a particular user or system can do and what AWS resources they might use.

IAM enables identity federation between your corporate directory and AWS services. This enables you to use your existing corporate identities to grant secure and direct access to AWS resources, such as Amazon S3 buckets, without creating a new AWS identity for those users.

IAM is a web service that enables AWS customers to manage users and user permissions under their AWS account.

For more information about this product, see [AWS Identity and Access Management \(IAM\)](#).

The objective of this connector is to support reading and provisioning of AWS IAM accounts, account groups and account group assignment.

Supported features

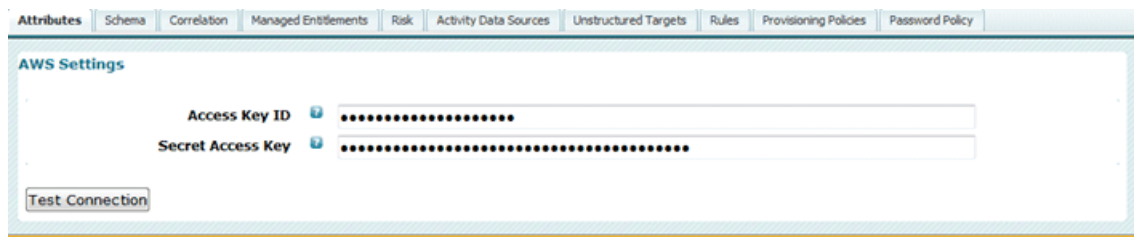
This release of the connector provides support for the following operations:

- Account Aggregation (Aggregates IAM Users under the AWS Account)
- Account-Group Aggregation (Aggregates IAM Groups under the AWS Account)
- Account Refresh
- Create/Update/Delete Account
- Create/Update/Delete Account-Group
- Account Enable (Activates ONLY ONE existing Access Key and Signing Certificate)
- Account Disable (Deactivates and/or deletes ALL existing Security Credentials)
- Reset Password (Does not require current password)
- Request/Remove Entitlement
- Direct Permissions on Account (Aggregation only)
- Direct Permissions on Account-Group (Aggregation only)

Pre-requisites

The connector requires the following Access Credentials to access the various IAM APIs:

- Access Key ID
- Secret Access Key



IAM is a feature of AWS account. If you are already signed up for a product that is integrated with IAM, you do not need to do anything else to sign up for IAM, and you will also not be charged extra for using it. You will be charged only for use of other AWS services by your users.

Note: IAM works only with AWS products that are integrated with IAM. For a list of such products, see [Integrating with Other AWS Products](#).

If you do not already have an AWS account, you need to create one to use IAM. You can create an AWS account when you sign up to use an AWS product for the first time. To sign up for AWS, perform the following:

1. Navigate to <http://aws.amazon.com>, and then click **Sign Up Now**.
2. Follow the on-screen instructions.
Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

An Access Key is automatically created upon creating an account. See the “Security Credentials” section of your account to obtain your Access Keys from the following link:

<http://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key>

Administrator permissions

Administrator should be configured to have proper access rights for modifying AWS IAM users.

Schema attributes

The following schema attributes are defined:

- Account schema
- Group schema
- Schema extension and custom attributes

Account schema

The following table lists the account schema:

| Name | Type | Description |
|---|--|--|
| UserName | String | The name identifying the user. |
| UserId | String | The stable and unique string identifying the user. |
| Path | String | Path to the user. |
| Arn | String | The Amazon Resource Name specifying the user. |
| CreateDate | String | The date when the user was created. |
| Access Keys <ul style="list-style-type: none"> ■ AccessKeyId ■ CreateDate ■ Status | List String String String | The Access keys associated with the user. The ID for this access key. The date when the access key was created. The status of the access key. |
| Signing Certificates <ul style="list-style-type: none"> ■ CertificateBody ■ CertificateId ■ Status ■ UploadDate | List String String String String | X.509 signing certificates associated with the user. The contents of the signing certificate. The ID for the signing certificate. The status of the signing certificate. The date when the signing certificate was uploaded. |
| Has Password | String | Whether the user password is set. |
| Multi-Factor Authentication Device <ul style="list-style-type: none"> ■ EnabledDate ■ SerialNumber | List String String | MFA device associated with the user. The date when the MFA device was enabled for the user. The serial number that uniquely identifies the MFA device. |
| Groups <ul style="list-style-type: none"> ■ GroupName | List String | The groups the user belongs to. The name that identifies the group. |
| Direct Permissions | List | The policies associated directly with the user as well as by virtue of groups the user belongs to. |

Group schema

The following table lists the group schema:

| Name | Type | Description |
|-------------------|--------|---|
| GroupName | String | The name that identifies the group. |
| GroupId | String | The stable and unique string identifying the group. |
| Path | String | Path to the group. |
| Arn | String | The Amazon Resource Name specifying the group. |
| CreateDate | String | The date when the group was created. |
| Group Permissions | List | The policies associated with the group. |

Schema extension and custom attributes

The connector handles all the attributes currently retrieved or provisioned by the respective IAM APIs at the time of designing and developing the connector. In addition, AWS IAM has fixed schemas and does not support adding custom attributes to any of the schemas. Therefore, the connector does not provide support for extending the schema and defining custom attributes.

Provisioning Policy attributes

The default provisioning policies are defined for Account and Account-Group.

Account

- **Create:** The following table lists the attributes that are required for creating an account.

| Name | Type | Required | Description |
|-----------|--------|----------|-------------------------------------|
| User Name | String | Yes | Name of the user to create. |
| Password | Secret | Yes | The new password for the user name. |
| Path | String | No | The path for the user name. |

- **Update:** The following table lists the attributes that are required for updating an account.

| Name | Type | Required | Description |
|---------------|--------|----------|------------------------|
| New User Name | String | No | New name for the user. |
| New Path | String | No | New path for the user. |

Account-Group

- **Create:** The following table lists the attributes that are required for creating a group.

| Name | Type | Required | Description |
|------------|--------|----------|------------------------------|
| Group Name | String | Yes | Name of the group to create. |
| Path | String | No | The path to the group. |

- **Update:** The following table lists the attributes that are required for updating a group.

| Name | Type | Required | Description |
|----------------|--------|----------|-------------------------|
| New Group Name | String | No | New name for the group. |
| New Path | String | No | New path for the group. |

Additional information

This section describes the additional information related to the AWS Connector.

Amazon Web Services Identity and Access Management API's

This section describes the API method used by the AWS IAM Connector.

Interaction with the application

The connector makes use of the REST requests to call the functionality exposed by an Amazon Web Services (AWS) API. REST or Query requests are simple HTTP or HTTPS requests that use an HTTP verb (such as GET or POST) and the Action or Operation parameter that specifies the API you are calling.

Calling an API using a REST or Query request is the most direct way to access a web service, but requires that your application handles low-level details such as generating the hash to sign the request and error handling.

The benefit of using a REST or Query request is that you have access to the complete functionality of an API. The connector makes use of the REST requests and has the provision to handle the low-level details.

APIs used

The following table lists the IdentityIQ operations along with the corresponding IAM APIs (Actions) used:

| IdentityIQ Operation | IAM API (Action) |
|----------------------|--------------------|
| Test Connection | ListAccountAliases |

Additional information

| | |
|--|---|
| Account Aggregation <ul style="list-style-type: none"> ■ Summary/Attributes ■ Access Keys ■ Signing Certificates ■ Password ■ Multi-Factor Authentication (MFA) Device ■ Entitlements/Groups ■ Direct Permissions | ListUsers ListAccessKeys ListSigningCertificates GetLoginProfile ListMFADevices ListGroupForUser ListUserPolicies, ListGroupPolicies |
| Account-Group Aggregation <ul style="list-style-type: none"> ■ Groups ■ Group Permissions | ListGroups ListGroupPolicies |
| Account Refresh <ul style="list-style-type: none"> ■ Summary/Attributes ■ Access Keys ■ Signing Certificates ■ Password ■ MFA Device ■ Entitlements/Groups ■ Direct Permissions | GetUser ListAccessKeys ListSigningCertificates GetLoginProfile ListMFADevices ListGroupForUser ListUserPolicies, ListGroupPolicies |
| Account Create <ul style="list-style-type: none"> ■ Set Password | CreateUser CreateLoginProfile |
| Account Update | UpdateUser |
| Account Delete <ul style="list-style-type: none"> ■ Read Entitlements/Groups ■ Remove Entitlements/Groups ■ Read Direct Permissions ■ Remove Direct Permissions ■ Read Security Credentials <ul style="list-style-type: none"> – Access Keys – Signing Certificates – Password – MFA Device ■ Remove Security Credentials <ul style="list-style-type: none"> – Access Keys – Signing Certificates – Password – MFA Device | DeleteUser ListGroupForUser RemoveUserFromGroup ListUserPolicies DeleteUserPolicy ListAccessKeys ListSigningCertificates GetLoginProfile ListMFADevices DeleteAccessKey DeleteSigningCertificate DeleteLoginProfile DeactivateMFADevice |
| Account-Group Create | CreateGroup |
| Account-Group Update | UpdateGroup |
| Account-Group Delete <ul style="list-style-type: none"> ■ Read Accounts in the Group ■ Remove Accounts from the Group ■ Read Group Permissions ■ Remove Group Permissions | DeleteGroup GetGroup RemoveUserFromGroup ListGroupPolicies DeleteGroupPolicy |
| Account Enable <ul style="list-style-type: none"> ■ Activate Access Keys (One only) ■ Activate Signing Certificates (One only) | UpdateAccessKey UpdateSigningCertificate |

| | |
|---|--|
| Account Disable <ul style="list-style-type: none"> ■ Deactivate Access Keys (All) ■ Deactivate Signing Certificates (All) ■ Delete Password ■ Deactivate MFA Device | UpdateAccessKey UpdateSigningCertificate DeleteLoginProfile DeactivateMFADevice |
| Reset Password | UpdateLoginProfile |
| Request Entitlement | AddUserToGroup |
| Remove Entitlement | RemoveUserFromGroup |

Troubleshooting

1 - Restore (Enable) security credentials

Restore security credentials for your IAM users.

CreateLoginProfile: Creates a password for the specified user, giving the user the ability to access AWS services through the AWS Management Console. IdentityIQ does not allow specifying the Password, which is a required parameter for this API, during **Account Enable** operation.

Workaround: The password must be set/created using Set/Reset Password operation to enable the account.

Chapter 12: SailPoint IdentityIQ Microsoft SharePoint Online Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 117 |
| Supported features | 117 |
| Prerequisites | 117 |
| Administrator permissions | 118 |
| Configuration parameters | 118 |
| Schema attributes | 119 |
| Account attributes | 119 |
| Group attributes | 119 |
| Provisioning Policy attributes | 120 |
| Install and register the IQService for Windows | 121 |
| Additional information | 121 |
| Unstructured Target Collector | 121 |

Overview

This connector manages the users, SharePoint groups and their attributes present on the Microsoft SharePoint Online directory. It does not manage the attributes associated with other products in Microsoft Office 365 suite like Office 365 Online directory store, Exchange Online, Lync Online.

The SharePoint online connector uses Windows Identity Foundation for the authentication and web services for managing users and groups to implement its functionalities in IQService which needs to be running on Windows 7 or Windows Server 2008 R2 computer.

Supported features

SailPoint IdentityIQ Microsoft SharePoint Online Connector provides support for the following features:

- Account Aggregation
 - Account-Group Aggregation
 - Account Refresh
 - Create/Delete/Update Account
 - Create/Delete Group
 - Add/Remove Entitlements
 - Target Aggregations for Sites, Lists, List Items, Folders, and Files
- For more information, see [“Unstructured Target Collector” on page 121](#).

Prerequisites

- The IQService must be installed on Windows 7 or Windows Server 2008 R2 computer.
The restriction for the Operating System version is applied due to the Office 365 cmdlets.

Configuration parameters

For more information, see “[Install and register the IQService for Windows](#)” on page 121.

- Windows Identity Foundation must be installed on the computer where IQService is installed. To install Windows Identity Foundation, download it from the following site:
<http://www.microsoft.com/en-us/download/details.aspx?id=17331>
- While creating user, SharePoint license is also assigned and it requires the Office 365 cmdlets to be present. This is a prerequisite for Office 365 and Exchange Online Connector. To install it perform the following steps:
 - Before installing the Office 365 cmdlets, install the Microsoft Online Services Sign-in Assistant if not already present on the system. Download and install one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 32 bit version](#)
 - [Microsoft Online Services Sign-In Assistant \(IDCRL7\) - 64 bit version](#)
 - To install the cmdlets, download one of the following from the Microsoft Download Center:
 - [Microsoft Online Services Module for Windows PowerShell \(32-bit version\)](#)
 - [Microsoft Online Services Module for Windows PowerShell \(64-bit version\)](#)
- .NET Framework 3.5.1 must be installed on the computer where IQService is installed.
- For target aggregation, redistributable package of SharePoint Foundation 2010 Client Object Model should be installed on the computer where IQService is present. It can be downloaded from the following site:
<http://www.microsoft.com/en-us/download/details.aspx?id=21786>

Administrator permissions

- Administrator should be a part of the Global Administrator role in Office 365. The administrator role can be changed from **Users=>Setting=>Assign Role=>Global Administrator**.
- The administrator should also be a Site Collection owner. The site collection owner can be assigned to the administrator from **Site Collections=>Select site which is to be managed=>Owners=>Manage Administrators=>Add the user to Site Collection Administrators**.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Microsoft SharePoint Online Connector uses the configuration parameters listed in the following table ([Table 26—Configuration parameters](#)):

Table 26— Configuration parameters

| Parameters | Description |
|----------------------|---|
| IQService Host* | Host name of the system where IQService is installed. |
| IQService Port* | Port number on which IQService is listening. Default: 5050. |
| Site Collection URL* | URL of SharePoint Online site to manage. |

Table 26— Configuration parameters

| Parameters | Description |
|---|--|
| Administrator User ID* | User ID or user principal name of the administrator. |
| Administrator Password* | Password of the administrator. |
| Page Size | The number of objects to fetch in a single page when iterating over large data sets. Default: 500. |
| * Indicates the mandatory attributes to create the application. | |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 27—Account attributes](#)):

Table 27—Account attributes

| Attributes | Description |
|---------------|--|
| LoginName | Login Name of the user. |
| UserName | User name of the user. |
| Name | Display Name of user. |
| Email | E-mail address of the user. |
| IsSiteAdmin | Specifies whether the user is a site collection administrator. |
| FirstName | First name of the user. |
| LastName | Last Name of the user. |
| PreferredName | Preferred name of the user. |
| ID | Member ID for the user. |
| WorkPhone | Office phone number of the user. |
| Groups | It specifies all the groups to which user belongs to. |

Group attributes

The following table lists the group attributes ([Table 28—Group attributes](#)):

Table 28—Group attributes

| Attributes | Description |
|-------------|--------------------------------|
| Name | Display name of the group. |
| Description | Description of the group. |
| OwnerID | Owner Id of the group. |
| ID | Identifier (ID) for the group. |
| OwnerIsUser | Owner type. |

Provisioning Policy attributes

Table 29—Provisioning Policy attributes for create, delete, and create group lists the provisioning policy attributes for create, delete, and Create Group respectively.

Table 29—Provisioning Policy attributes for create, delete, and create group

| Attributes | Description | Required attribute |
|--|--|---|
| Provisioning policy attributes for create | | |
| UserName | User Name of the User. | Yes |
| Email | Email of the user. | No |
| AccountSkuld | Licensing plans that are available for SharePoint (AccountSkuld). These plans can be retrieved by executing the powershell cmdlet Get-MsolAccountSku and it will be displayed in domain:plan format under the AccountSkuld column. Note: Before provisioning, replace/modify the default values of 'AccountSkuld' in provisioning policy with the values retrieved as mentioned above. | Required only if multiple licensing plans are available. |
| Provisioning policy attributes for delete | | |
| RemoveSharePointLicense | Whether or not the SharePoint license should be removed after deleting the account. | No |
| AccountSkuld | Licensing plans that are available for SharePoint (AccountSkuld). These plans can be retrieved by executing the powershell cmdlet Get-MsolAccountSku and it will be displayed in domain:plan format under the AccountSkuld column. Note: Before provisioning, replace/modify the default values of 'AccountSkuld' in provisioning policy with the values retrieved as mentioned above. | Required only if RemoveSharePointLicense attribute is true and there are multiple licensing plans available. |
| Provisioning policy attributes for create group | | |
| Name | Name of the group. | |
| DefaultUser | Default user of the group. | |
| Permission | Group permission. | |

Table 29—Provisioning Policy attributes for create, delete, and create group (Continued)

| Attributes | Description | Required attribute |
|-------------|---------------------------|--------------------|
| Owner | Owner of the group. | |
| OwnerIsUser | Owner type. | |
| Description | Description of the group. | |

Install and register the IQService for Windows

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the **IQService.zip** archive from the **IIQHOME\WEB-INF\bin\win** directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.

IQService -i

It registers the service with the new registry path, **HKEY_LOCAL_MACHINE\SOFTWARE\SailPoint\IQService** with the following keys:

- **port**: port to listen
- **tracefile**: path to the tracefile
- **tracelevel**: 0 (off)
3 (verbose)

4. Start the service either from the Services Applet or from the command line by running the following command:

IQService -s

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information

Additional information

This section describes the additional information related to the Microsoft SharePoint Online Connector.

Unstructured Target Collector

SharePoint Online uses a data structure which requires the configuration of the **Unstructured Targets** tab to collect targeted data and correlates it with **LoginName** for Accounts and **Name** for groups. For more information on the Unstructured Targets Tab, see “Unstructured Targets Tab” section of the *SailPoint IdentityIQ User’s Guide*.

Additional information

SharePoint Online Target Collector supports aggregation for Sites, Lists, List Items, Folders and Files. The objects can be filtered based on various filters configured on the Unstructured Targets Tab.

| Attribute | Description | Possible values |
|-------------------------------|--|--|
| Site Collection URL | URL of Site or Site Collection for target aggregation. | URL. Cannot be blank |
| UserName | User to be used for aggregating the site targets. | admin@domain.onmicrosoft.com Cannot be blank. |
| Password | Password for UserName | |
| Target Types Filter | As mentioned above, the Target Collector supports aggregating Sites, Lists, List Items and Files. Using this filter, any of these target types can be selectively aggregated. | Any combination of following separated by comma: Sites,Files,Lists,ListItems,Folders,Files ListItem specific filtration - for example, Document,Discussion,Picture,Wiki Page and so on If not specified, all target types would be aggregated. Default – Not specified |
| Include inherited permissions | SharePoint has hierarchy structure for targets. The child target can inherit permissions from the parent. In that case, the permissions of child and parent would be same. For example, all files in a folder can inherit permissions from folder. Hence aggregating file permissions may not be of interest. This filter can include or exclude such targets. | True/False If true, the target aggregation will fetch all targets including the one having inherited permissions. Default: True |
| Site Filter Type | This is used in combination to the Site Filter. This tells whether the Site Filter define the inclusion filter or exclusion filter. | Include/Exclude Default: Include |
| Site Filter | Targets with path containing Words / phrases mentioned here can be selectively included or excluded depending on the Site Filter Type | Words/phrases separated by comma. If not specified all the targets would be aggregated. Default: Not specified |

Chapter 13: SailPoint IdentityIQ NetSuite Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 123 |
| Supported features | 123 |
| Supported Managed Systems | 124 |
| Administrator permissions | 124 |
| Configuration parameters | 125 |
| Schema attributes | 125 |
| Account attributes | 125 |
| Group attributes | 126 |
| Schema extension and custom attributes | 126 |
| Provisioning Policy attributes | 127 |
| Additional information | 128 |
| NetSuite Application Program Interface (API) | 128 |

Overview

NetSuite is cloud-based Software-as-a-Service integrated business management software. NetSuite's cloud business management system includes ERP/accounting, order management/inventory, CRM, Professional Services Automation (PSA) and E-commerce.

Enterprise Resource Planning (ERP) in NetSuite encompasses several areas of your business, including accounting, inventory, order management, project management, and employee management.

For more information, see <http://www.netsuite.com/portal/products/main.shtml>

NetSuite Connector will manage the employee data in the NetSuite ERP system. The connector is a write-capable connector which manage the following entities:

- Employee Account
- Employee Role
- Employee Entitlement

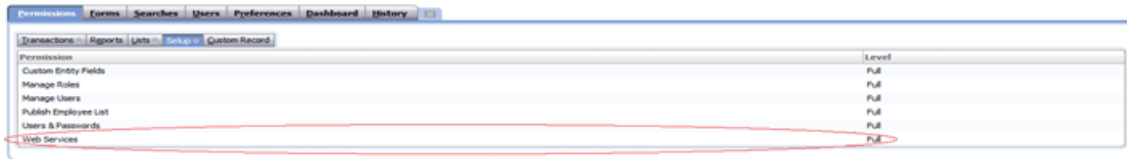
Supported features

This release of the connector provides support for the following operations:

- Account Aggregation
- Group Aggregation
- Refresh Account
- Create/Delete Account
- Add/Delete Account Entitlement
- Enable/Disable Account
- Change Password
- Pass through Authentication

Overview

Note: For Pass through Authentication, the account should have at least one role assigned with permissions required to perform the operation. Also this role needs to be Web Service enabled role as displayed in the following figure:



The screenshot shows the NetSuite 'Permissions' page. The 'Web Services' permission is highlighted with a red oval, and its 'Level' is 'Full'.

| Permission | Level |
|-----------------------|-------|
| Custom Entity Fields | Full |
| Manage Roles | Full |
| Manage Users | Full |
| Publish Employee List | Full |
| Users & Passwords | Full |
| Web Services | Full |

Supported Managed Systems

SailPoint IdentityIQ NetSuite Connector supports the following managed system:

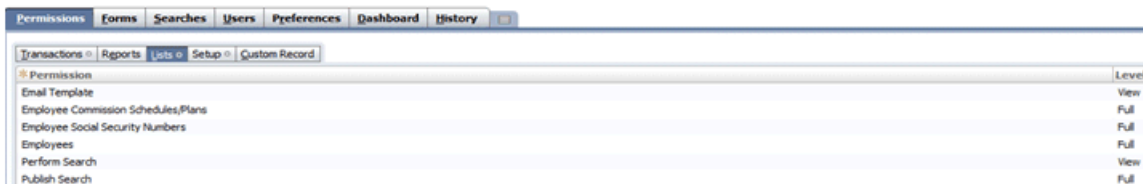
- Netsuite 2012_1

Administrator permissions

The NetSuite Connector administrator must be able to perform the following operations on NetSuite employee data:

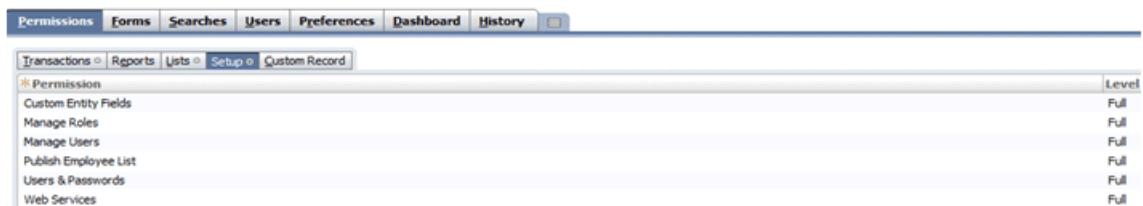
- Search
- Create
- Update
- Delete
- Access Custom Attributes

Hence a role is required which has the permissions to the above operations. We need to create a role in NetSuite.



The screenshot shows the NetSuite 'Permissions' page with the following permissions and levels:

| Permission | Level |
|-------------------------------------|-------|
| Email Template | View |
| Employee Commission Schedules/Plans | Full |
| Employee Social Security Numbers | Full |
| Employees | Full |
| Perform Search | View |
| Publish Search | Full |



The screenshot shows the NetSuite 'Permissions' page with the following permissions and levels:

| Permission | Level |
|-----------------------|-------|
| Custom Entity Fields | Full |
| Manage Roles | Full |
| Manage Users | Full |
| Publish Employee List | Full |
| Users & Passwords | Full |
| Web Services | Full |

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The NetSuite connector uses the following connection parameters:

| Parameters | Description |
|-------------------------|--|
| Account ID* | the account number assigned to an organization by NetSuite. This account number must be provided by each login request. This can be found by navigating to Setup => Integration => Web Services Preferences . |
| Role ID | When logging in using Web Services provide a role id along with your credentials. The role defined here must be a valid role contained in the Employee record of the given user. If no role id is provided, then the user's default role is used. If neither the request nor the Web Services default role is set, then the user's default UI role is used, provided it has the Web Services permission. For security reasons, it is recommended that you restrict permissions levels and access allowing only the most restricted permissions necessary to perform a given set of operations. For more information about the permissions, see “Administrator permissions” on page 124 . |
| Administrator Email* | Email of the Account in Employee package having provisioning privileges. |
| Administrator Password* | Password of the employee Account. |
| Page Size | Limit to fetch number of accounts or groups per iteration through NetSuite Connector. If the value is not set then the default value is 50. |

Schema attributes

The following schema attributes are defined:

- Account schema
- Group schema
- Custom attributes

Account attributes

The following table lists the account schema:

| Attribute Name | Description |
|----------------|--|
| EmpID | Employee ID |
| InternalID | Auto generated Internal ID of the employee |
| EmployeeStatus | The status of employee |
| Email | Email ID of employee |
| Initial | The initials of first name and last name |

Schema attributes

| | |
|--------------------------|---|
| OfficePhoneNumber | Office phone number of employee |
| HomePhoneNumber | Home phone number of employee |
| MobilePhoneNumber | Mobile number of employee |
| Department | Department of employee |
| Class | Class of employee |
| BillingClass | Billling class of employee |
| Groups (Entitlements) | Groups associated to the employee |
| GlobalSubscriptionStatus | Subscription status of employee |
| SocialSecurityNumber | Security number of employee |
| Supervisor | Supervisor of employee |
| DateOfHiring | Date of hiring of employee |
| Type | Working type of employee |
| JobTitle | Job title of employee |
| DateOfBirth | Date of birth of employee |
| JobDescription | Description of job of employee |
| TimeApprover | Approver of time for the employee (some one like supervisor or manager) |

Group attributes

The following table lists the group schema:

| Attribute Name | Description |
|-----------------|---|
| GroupName | Name of the group |
| GroupInternalID | Auto generated Internal id of the group |

Schema extension and custom attributes

NetSuite system allows the support for extending the schema through custom entity fields. Custom entity fields are fields that you can add to your entity records to gather information specific to your business needs. Entity custom fields can be added to existing and custom sub tabs on the entry forms you use to enter entity records in your NetSuite account.

NetSuite connector supports the read and write of custom attributes.

Following NetSuite Custom field type are supported in IdentityIQ

- Check Box
- Date
- Free-Form Text
- Email Address
- Phone Number
- HyperLink

Supporting of custom attributes

Perform the following to support the custom attributes from IdentityIQ:

- Add the custom attribute name in the schema by clicking **Add attribute** button.
- Add the following lines in the application debug page:

```
<entry key = "customAttribute" >
  <value>
    <List>
      <String>custom1</String>
      <String>custom2</String>
    </List>
  </value>
</entry>
```

Note: No code change would be required while adding new custom attributes in schema. This is applicable only for custom attributes.

Provisioning Policy attributes

The NetSuite connector is pre-configured with an account creation provisioning policy that includes the commonly-used attributes that need to be set when creating an account. This field list can be modified as required.

The attributes listed in the following table are required for creating an user.

| Attribute Name | Description |
|--------------------|--------------------------------------|
| EmpID (Entity ID)* | Employee name |
| *password* | Password for the employee |
| Email* | Email of the employee |
| OfficePhoneNumber | Office phone number for the employee |
| Fax | Fax for the employee |

In the above table, **EmpID** is the minimum parameter which is required to create a user on NetSuite server. But in IdentityIQ a user can only be created after assigning a role to it.

In NetSuite when a role is assigned to a user, the user requires UserName, Email and password as mandatory parameter for accessing the NetSuite server.

Additional information

Note: The field list can also be extended by adding custom attributes provided the attributes are defined in the application schema. For more information, see [“Schema extension and custom attributes”](#) on page 126.

Additional information

This section describes the additional information related to the NetSuite Connector.

NetSuite Application Program Interface (API)

SuiteTalk exposes NetSuite as a data source for programmatic access. The following operations supported in SuiteTalk would be used by NetSuite Connector

| Operation/API | Summary |
|-----------------------|---|
| add | Use to add record into the system. The system returns a NetSuite identifier (internalId) that is unique for each record created within a record type. |
| changePasswordOrEmail | Use to change a user's email or password |
| get | Use to query the system for one record. You must provide either the internal or external ID and the record type for each query item. |
| getCustomizationId | Use to retrieve the internalIds, externalIds, and/or scriptIds of all custom objects of a specified type. |
| login | Use to login into NetSuite. This operation is similar to the NetSuite UI and requires you to provide a valid username, password, role, and account number. |
| logout | Use to logout from the system. The logout operation invalidates the current session. |
| search | Use to search for a set of records based on specific search criteria. This operation supports pagination, so that large result sets can be retrieved in smaller sets. |
| searchMore | Used to retrieve more records after an initial search operation is invoked. |
| update | Use to update existing record in the system by providing new values for the fields to be updated for each record. The records to be updated are identified by either the internal or external ID and the record type. |
| delete | Use to delete an existing record in the system by providing the internal id and record type. |

Note: For more information, see *NetSuite SuiteTalk (Web Services) Platform Guide*.

Chapter 14: SailPoint IdentityIQ JDBC Connector

The following topics are discussed in this chapter:

| | |
|---------------------------------|-----|
| Overview | 129 |
| Supported features | 129 |
| Supported Managed Systems | 130 |
| Pre-requisites | 130 |
| Administrator permissions | 130 |
| Configuration parameters | 130 |
| Schema Attributes | 131 |

Overview

The JDBC Connector is used for Read/Write operations on the data of JDBC enabled database engines. This connector supports flat table data. To handle complex, multi-table data, you need to define a rule and a more complex SQL statement.

This connector can be configured to enable the automatic discovery of schema attributes. See [“Schema Attributes” on page 131](#).

IdentityIQ supports for the following additional JDBC Connector features in version 5.2 and later:

- Ability to provide the SQL statement or stored procedure during application configuration for automatic discovery of account-group schema attributes from same or different database used for the account schema.
- Ability to define provisioning rule(s) called for each row in the data file to provision account and group attributes.
- Ability to define separate provisioning rule for specific operation called for each row in the data file to provision account and group attributes. Operation that include are Enable, Disable, Unlock, Delete, Create, and Modify.

Note: An example of a provisioning rule is located in `examplerules.xml` file.

Supported features

SailPoint IdentityIQ JDBC Connector provides support for the following features:

- Account Aggregation
- Group Aggregation
- Refresh Account
- Create/Delete
- Add /Delete Account Entitlement
- Enable/Disable Account
- Change Password

Supported Managed Systems

SailPoint IdentityIQ JDBC Connector supports the following Managed System:

- Any database having JDBC Driver. For example, MySQL, Oracle, DB2, SQLServer and Sybase

Pre-requisites

An appropriate JDBC driver for the database.

Administrator permissions

IdentityIQ application would communicate to a database user who has permission to do get and set operation on the tables.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The JDBC connector uses the following connection attributes:

| Attribute | Description |
|-------------------------|--|
| user | The user with which to connect to the host. |
| password | The password associated with the specified user. |
| url | The URL with which to connect to the database. |
| driverClass | The Java JDBC class to use for the connection. |
| SQL | The SQL attribute can be used to customize the select statement that is generated when iterating over objects. You can specify the exact SQL that is executed if you want to filter out objects or only want to select a few objects from a table. Additionally, if you need to perform joins between more than one table, it's impossible to describe with the schema alone. By default if the SQL option is null when the query string is built using the schema attributes and nativeObjectType. |
| buildMapRule | The rule called for each row returned by the database after the SQL has been executed. The rule uses ResultSet and builds a Map out of it to be consumed by IdentityIQ. |
| mapToResourceObjectRule | Rule that is called to override the transformation of the data from the Map<String,String> form into a ResourceObject . |
| mergeRows | true or false. Indicates if the connector needs to be aware of rows that are alike so they can be merged. |
| indexColumns | The column name that indicates how the like rows are correlated. |
| mergeColumns | The columns that are used if the default merge implementation is true |

| Attribute | Description |
|---------------|---|
| mergeMapsRule | A rule can be called to merge objects. This overrides the default implementation. |

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

The JDBC connector's most important attribute is the SQL statement. In many cases this is a stored procedure. (**call mystoredProcedure**). In other cases it is select from a table with any number of joins included. If this connector is configured to use the automatic discovery function, it connects to the database and executes the statement provided and then uses the meta-data returned from the result to build the column names.

JDBC Connector - Merging and Ordering

Starting in version 5.0, IdentityIQ checks the order of the data returned from the database when merging to prevent data loss. When merging, it is very important to have the ORDER BY clause in your SQL statement to prevent out of order errors.

Schema Attributes

Chapter 15: SailPoint IdentityIQ PeopleSoft Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 133 |
| Supported features | 133 |
| Supported Managed Systems | 133 |
| Pre-requisites | 133 |
| Administrator permission | 134 |
| Configuration parameters | 134 |
| Schema attributes | 134 |
| Account attributes | 134 |
| Group attributes | 135 |
| Additional information | 136 |
| Create the Component Interfaces | 136 |
| Create and Copy the required jar files | 136 |
| Configure Component Interface Security | 137 |

Overview

The PeopleSoft Connector manages the administrative entities of PeopleSoft server (User Profiles and Roles). The PeopleSoft connector communicates to the PeopleSoft server through component interfaces.

Supported features

The PeopleSoft Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Create/Update/Delete Account
- Get/Sync Account
- Enable/Disable Account
- Change Password
- Discover Schema

Supported Managed Systems

SailPoint IdentityIQ PeopleSoft Connector supports the following managed systems:

- PeopleTool version 8.48, 8.49, 8.50, 8.51, and 8.52

Pre-requisites

To use the PeopleSoft connector, you must first configure the component interfaces on PeopleSoft. This requires the following steps:

1. [Create the Component Interfaces](#)
2. [Create and Copy the required jar files](#)

Configuration parameters

3. [Configure Component Interface Security](#)

Administrator permission

The PeopleSoft Administrator must have the **IIQ_ROLE** role created in the above section for the proper functioning of the connector.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The PeopleSoft connector uses the following connection attributes:

| Attribute | Description |
|--------------------------|---|
| host | The hostname of the PeopleSoft server. |
| port | The port on which the PeopleSoft server is listening. |
| user | The user name used to login to PeopleSoft. |
| password | The password to use to login to PeopleSoft. |
| componentInterface | The name of the PeopleSoft component interface to use to read accounts. |
| group.componentInterface | The name of the PeopleSoft component interface to use to read groups. |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|-------------------|---|
| UserId | The PeopleSoft User ID. |
| AccountLocked | Status of Account if it is locked or not. |
| AlternateUserID | User ID Alias. |
| CurrencyCode | Currency code of the user. |
| DefaultMobilePage | Default mobile page. |
| EffectiveDateFrom | Workflow attribute - from date. |

| Attributes | Description |
|------------------------------|--|
| EffectiveDateTo | Workflow attribute - to date. |
| EmailAddresses | Email address of the user. |
| EmailUser | Routing preferences - email user. |
| ExpertEntry | Enable expert entry. |
| FailedLogins | Number of failed logins. |
| IDTypes | User ID types and values. |
| LanguageCode | Language code. |
| LastUpdateDateTime | Last update date/time. |
| LastUpdateUserID | Last update user ID. |
| MultiLanguageEnabled | Multi-language enabled. |
| NavigatorHomePermissionList | Default navigator home page permission list. |
| Opertype | Use external authentication. |
| PasswordExpired | Is password expired. |
| PrimaryEmailAddress | Primary email address. |
| PrimaryPermissionList | Primary permission list. |
| ProcessProfilePermissionList | Process profile permission list. |
| roleNames | Roles assigned to the user. |
| RowSecurityPermissionList | Row security permission list. |
| SymbolicID | Used to map the User Id to Access ID. |
| UserDescription | Description of the user. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|---------------------|--|
| ROLENAME | Name of the role. |
| ROLETYPE | Type of the role. |
| RolePermissionLists | Permission List for the role. |
| DESCR | Description of the role. |
| DESCRLONG | Long description. |
| ALLOWNOTIFY | Workflow routing - allow notifications. |
| ALLOWLOOKUP | Workflow routing - allow recipient lookup. |
| LASTUPDDTTM | Last update date/time. |
| LASTUPDOPERID | Last update user ID. |

Additional information

| Attributes | Description |
|---------------------------|---|
| Roles that can be granted | Roles that can be granted by this role. |
| Roles that can grant | Roles that can grant this role. |

Additional information

This section describes the additional information related to the PeopleSoft Connector.

Create the Component Interfaces

1. Login to the PeopleTools Application Designer.
2. Unzip the **tdi.zip** file in **integration/TDI** to a temporary directory.
3. Select **Tools ==> Copy Project ==> From File...** and browse to the temporary directory from step 2. Navigate to **config/applications/peoplesoft** and highlight the **IIQ_CONN** directory.
4. Select the **IIQ_CONN** project to open it.
5. Highlight Component Interfaces and click **Copy** to copy the project into PeopleSoft.

Create and Copy the required jar files

The following jars must be copied from the PeopleSoft server:

- iiqPeopleSoftComplnt.jar
- psjoa.jar

The **iiqPeopleSoftComplnt.jar** file contains the PeopleSoft Component Interface java classes. It must be generated from the respective PeopleSoft resource and then copied into the IdentityIQ classpath.

Perform the following steps to create the **iiqPeopleSoftComplnt.jar** file from the Component interface java files.

1. Login to PeopleSoft Application Designer in two tier mode.
2. Open the **IIQ_CONN** Component Interface project and open all the component interfaces by double clicking each component interface.
3. From the menu select **Build ==> PeopleSoft APIs**.
4. In the JAVA Classes frame check Build and select the appropriate Component Interfaces from the drop down menu. You must select the following options from the drop down menu:
5. From the Build PeopleSoft API Bindings window, select the JAVA classes Build check box and clear the COM Type Library and C Header Files Build check boxes.
 - Complntfc.ComplntfcPropertyInfo
 - Complntfc.ComplntfcPropertyInfoCollection
 - PeopleSoft.* (all Component Interfaces that begin with the prefix PeopleSoft)
 - Complntfc.IIQ_* (all Component Interfaces that begin with the prefix Complntfc.IIQ_)

Note: If you need to generate Component Interface Java files for the entire group of Component Interfaces click **ALL**.

Specify the appropriate file path for the JAVA files. The Component Interface JAVA files are generated in the **PeopleSoft\Generated\CompIntfc** directory that is created in the specified location. For example, if you specify **C:\CI** as the file path, then the Component Interface Java files are generated in **C:\CI\PeopleSoft\Generated\CompIntfc**.

6. Compile the JAVA files by performing the following steps:
 - a. Open the command prompt and change directories to the folder where the generated JAVA files are located. For example, **C:\CI**.
 - b. Navigate to the **PeopleSoft\Generated\CompIntfc** directory.
 - c. Run the following command:

```
javac -classpath %PS_HOME%\class\psjoa.jar *.java
```

Where %PS_HOME% is the location that PeopleSoft is installed.

Important: Ensure that the JAVA compiler used for compiling the generated JAVA files is compatible with the JAVA provided with the PeopleSoft installation that needs to be managed.

- d. (Optional) You can delete all the generated java files from the existing directory, however, do not delete the **.class** files.
7. Perform the following steps to package the compiled files as the **iiqPeopleSoftCompInt.jar** file:
 - a. Open the Command prompt and change directories to the folder where the generated JAVA files are located. For example **cd C:\CI**
 - b. Run the command: **jar -cvf iiqPeopleSoftCompInt.jar ***
8. Copy the generated **iiqPeopleSoftCompInt.jar** file into the IdentityIQ classpath.
9. Copy **psjoa.jar** from **%PS_HOME%\classes** into the IdentityIQ classpath.

Configure Component Interface Security

Before using the connector, you must allow the PeopleSoft user, for whom the connector is configured, to access the generated component interfaces.

To set security for the PeopleTools project, perform the following:

1. Log into the PeopleSoft web interface.
2. Navigate to **PeopleTools ==> Security ==> Permissions & Roles ==> Permission Lists**.
3. Click **Add a New Value** to create a new permission list. Type **IIQ_ALL** as the name of the permission list, then click **Add**.

Additional information

4. Click the Component Interfaces tab and add the following to the list:
 - IIQ_CURCODE
 - IIQ_DEL_ROLE
 - IIQ_DEL_USER
 - IIQ_IDTYPE
 - IIQ_LANG
 - IIQ_PERMLIST
 - IIQ_ROLES
 - IIQ_USERS
5. For each added component interface, click **Edit ==> Full Access (All)**, then click **OK**.
6. Click **Save** to save the new permission list.
7. Navigate to **PeopleTools ==> Security ==> Permissions & Roles ==> Roles**.
8. Click **Add a New Value** to create a new role. Type **IIQ_ROLE** as the name then click Add.
9. Type **Allows access to the IdentityIQ component interfaces** as the description.
10. Click the **Permission Lists** tab and add the **IIQ_ALL** permission list. Click **Save** to save the role.
11. Navigate to **PeopleTools ==> Security ==> User Profiles**, and select the user that is being used in the connector.
12. Click the **Roles** tab and add the **IIQ_ROLE** role. Click **Save** to add the role to the user.

Chapter 16: SailPoint IdentityIQ Siebel Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 139 |
| Supported features | 139 |
| Supported Managed Systems | 140 |
| Pre-requisites | 140 |
| Administrator permission | 140 |
| Configuration parameters | 140 |
| Schema attributes | 141 |
| Account attributes | 141 |
| Account Group attributes | 142 |
| Adding new custom attributes in schema | 143 |
| Provisioning policy attributes | 143 |
| Troubleshooting | 144 |

Overview

The Siebel Connector manages entities in Oracle's Siebel CRM. Here **Employee** is managed as Accounts and **Position** as Account Groups. By default, the Siebel Connector uses the Employee Siebel business component of the Employee Siebel business object for account provisioning. For Account Group provisioning Position business component of Position business object is used by Connector. However, the Connector can be configured to manage other Siebel Business Object/Component in the Account/Account Group provisioning. The Connector manages both single and multi-valued attributes of Siebel system. The Connector schema can be modified to manage attributes other than Schema that comes by default with Connector.

Supported features

SailPoint IdentityIQ Siebel Connector supports the following functions:

- Account Aggregation
- Account-Group Aggregation
- Create/Update/Delete Account
- Get/Sync Account
- Enable/Disable Account
- Change Password
- Create/Update/Delete Account-Group
- Add/Remove entitlement

The Account Aggregation and Get/Sync Account operations will retrieve and display the status (Enabled or Disabled) of the Account. The Entitlements will also be retrieved during these operations.

In Enable Account operation the **Employment Status** attribute is set to **Active** in Siebel system. The **Employment Status** attribute is set to **Terminated** in Disable Account operation.

Supported Managed Systems

SailPoint IdentityIQ Siebel connector supports the following Managed Systems:

- Siebel CRM version 8.2

Pre-requisites

Following Siebel JAR files are required in the **WEB-INF/lib** directory of the IdentityIQ instance:

- **Siebel 7.8 through 8.2:** **Siebel.jar** and **SiebelJI_<<Language>>.jar**
- **Siebel 7.5 through 7.7:** **SiebelJI_Common.jar**, **SiebelJI_<<Language>>.jar**, and **SiebelJI.jar**

For example, for Siebel CRM 8.2 with English language: **Siebel.jar**, **SiebelJI_enu.jar**

The Siebel JAR files are available in the **SIEBEL_INSTALLATION_DIRECTORY/siebsrvr/CLASSES** directory.

Note: Do not copy JAR files for multiple versions of Siebel into the **WEB-INF/lib** directory; it may create conflicts at runtime.

Note: - Siebel Connector requires JRE 1.6 to manage Siebel CRM 8.2
- Siebel Connector requires JRE 1.5 to manage Siebel CRM 8.0
Only one version of Siebel CRM can be managed using one IdentityIQ instance.

Administrator permission

The Siebel Connector requires Siebel administrator credentials to accomplish provisioning tasks. The administrator user name and password configured for the Siebel connector must be assigned sufficient privileges within Siebel to create new records and to update existing records for the specified business component.

For example, SADMIN user which is created during Siebel server installation is one of the example of administrator.

Note: A responsibility named “Siebel Administrator” assigned to this user gives access to all views.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Siebel Connector uses the connection parameters listed in the following table:

Table 30—Configuration parameters

| Parameter | Description |
|--------------------|---|
| Transport Protocol | Transport protocol while communicating with Siebel server. Select TCPIP or NONE. Default: TCPIP |
| Encryption | Data Encryption method. Select RSA or NONE. Default: NONE |
| Compression | Data Compression technique. Select ZLIB or NONE. Default: ZLIB |
| Siebel Server Host | Host Name where Siebel server is installed. |

Table 30—Configuration parameters

| Parameter | Description |
|--------------------------------|---|
| SCB Port | Listening port number for the Siebel Connection Broker (alias SCBroker). Sample value : 2321 |
| Siebel Enterprise Name | Name of Siebel Enterprise. Sample value: SBA_82 |
| Siebel Object Manager | Name of Siebel Application Object Manager. Sample value: SCCObjMgr |
| Admin User Name | User ID of the target system user account that you want to use for connector operations. Sample value: SADMIN |
| Password | Password of the target system user account that you want to use for connector operations. Sample value: sadmin |
| Language | Language in which the text on the UI is displayed. Specify any one of the following values: <ul style="list-style-type: none"> ■ For English: ENU ■ For Brazilian Portuguese: PTB ■ For French: FRA ■ For German: DEU ■ For Italian: ITA ■ For Japanese: JPN ■ For Korean: KOR ■ For Simplified Chinese: CHS ■ For Spanish: ESP ■ For Traditional Chinese: CHT |
| Account Business Object | Business Object for Account. Default value: Employee |
| Account Business Component | Business Component for Account. Default value: Employee |
| Entitlement Business Object | Business Object for Entitlement. Default value: Position |
| Entitlement Business Component | Business Component for Entitlement. Default value: Position |
| Siebel URL | Siebel server connection string. The server is connected using connection string. Specific parameters defined in the form are ignored. For example: <code>siebel.transport.encrypted.compression://host:port/EnterpriseServer/AppObjMgr_lang" lang="lang_code"</code> |

Schema attributes

By default the following mentioned set of attributes are managed:

Account attributes

The following table lists the account attributes (Siebel **Employee** attributes):

Schema attributes

| Attributes | Description |
|---------------------------|---|
| Login Name | Employee's login name. |
| First Name | Employee's first name. |
| Last Name | Employee's Last name. |
| Position | Multi-value attribute that contains a list of all positions assigned to employee. |
| Primary Position | Employee's primary position. |
| Responsibility | Multi-value attribute that contains a list of all responsibilities of employee. |
| Primary Responsibility Id | Employee's Primary responsibility ID. |
| Division | Division |
| Employment Status | Employment Status |
| Street Address | Street Address |
| Job Title | Job Title |
| Phone Number | Phone Number |
| Fax Number | Fax Number |
| Hire Date | Hire date |
| Alias | Alias |
| State | State |
| Availability Status | Availability status of employee. |
| ManagerLogin | Employee's Manager login. |

Account Group attributes

The following table lists the Account Group attributes (Siebel **Position** attributes):

| Attributes | Description |
|----------------------|--|
| Id | Unique Id for Position Entity. |
| Name | Name of Position. |
| Last Name | Last Name of Employees having this Position. |
| Division | Division of Position. |
| Role | Role |
| Start Date | Start date for allocation of Position to Employee referred by Last Name. |
| Position Type | Position Type. |
| Parent Position Name | Parent Position's name. |

Note: The search is made on *identityAttribute* while finding records. By default, "Login Name" for Account and "Id" for Account Group is set in the *identityAttribute*.

Adding new custom attributes in schema

Currently Siebel Connector schema provides basic minimum attributes required to manage Employee and position. If you want to enhance schema, you can add more attributes to the existing schema. You can use Siebel Tools to get the details about attributes to be managed using schema. If you add any new multi value attribute, configure the following attribute in Application using the debug page:

```
<entry key="customMVGAttr">
  <value>
    <List>
<!-- Format is <<Multi value attribute Name>>:<<MVG Business component>>:<<Business Object for field>>:<<Business component for field>>:<<Search key for multi value field>> -- >
      <String>Position:Position:Position:Position:Id</String>
      <String>Responsibility:Responsibility:Responsibility:Responsibility:Name</String>
    </List>
  </value>
</entry>
```

Note: As position and responsibility are main multi value field in Employee, if you do not configure it, Siebel Connector will assume the default business components and objects. But for other Multi value attribute to work, you need to configure this attribute in Application.

Provisioning policy attributes

The following table lists the provisioning policy attributes for Create and Update of Accounts and Group:

| Attributes | Description |
|-----------------------|---|
| Create Account | |
| Login Name | Employee's login name. |
| First Name | Employee's first name. |
| Last Name | Employee's last name. |
| Position | Multi-value attribute that contains a list of all positions assigned to employee. |
| Primary Position Id | Employee's primary position Id. |
| Responsibility | Multi-value attribute that contains a list of all responsibilities of employee. |
| Password | Employee account password. |
| Verify Password | Employee account password. |
| Job Title | Job title. |
| Employee Type | Employee type. |
| Update Account | |
| First Name | Employee's first name. |
| Last Name | Employee's last name. |
| Responsibility | Multi-value attribute that contains a list of all responsibilities of employee. |
| Primary Position Id | Employee's primary position Id. |

Troubleshooting

| Attributes | Description |
|---------------------|-----------------------|
| Create Group | |
| Position | Name of position. |
| Division | Division of position. |
| Position Type | Position type. |
| Parent Position Id | Parent position's Id. |
| Update Group | |
| Position Type | Position type. |
| Parent Position Id | Parent position's Id. |

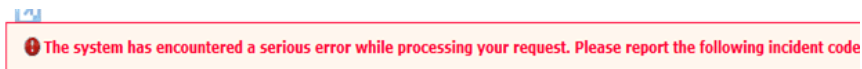
Troubleshooting

1 - When Siebel JAR files are not copied correctly in the WEB-INF/lib directory of the IdentityIQ instance, the following errors are obtained:

- Test connection fails with the following error:



- During add new entitlement the following error message is displayed:



Resolution: Copy the correct Siebel JAR files. For more information, see the [“Troubleshooting”](#) on page 144.

Chapter 17: SailPoint IdentityIQ Lotus Domino Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 145 |
| Supported features | 145 |
| Supported Managed Systems | 146 |
| Pre-requisites | 146 |
| Administrator permissions | 147 |
| Configuration parameters | 147 |
| Schema attributes | 148 |
| Account attributes | 148 |
| Group attributes | 150 |
| Provisioning policy attributes | 150 |
| Create account attributes | 150 |
| Create group attributes | 152 |
| Update policies | 152 |
| Install and register the IQService | 154 |
| Additional information | 154 |
| ID Vault functionalities | 155 |
| Password management | 155 |
| Troubleshooting | 155 |

Overview

SailPoint IdentityIQ Lotus Domino Connector was developed to manage the accounts and groups contained in a Notes database.

Supported features

SailPoint IdentityIQ Lotus Domino Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Create\Delete\Refresh Account
- Update Account (Update attributes, Rename, Recertify, Move user to a different certifier)
- Create\Update\Delete Account-Group
- Add\Remove Entitlement
- Enable\Disable\Unlock Account
- Change Password (HTTP - Default and ID file)
For more information, see [“Password management” on page 155](#).
- Authenticate (using HTTP password only)

Overview

- ID Vault functionalities:
 - Reset Password
 - Extract ID from vault
 - Upload ID to vault
 - Sync ID file

Supported Managed Systems

- Domino Server 8.0 and above
- Domino Server 8.5 and above (for “ID Vault functionalities”)
- Domino Server 9.0 and above (for “ID Vault functionalities”)

Pre-requisites

- The computer running IdentityIQ should have the **NCSO.jar** file in the classpath.
- Ensure that Domino server **notes.ini** file contains the following line:
ServerTasks=<any other tasks>, DIIOP, HTTP
HTTP task is required to be mentioned only if the DIIOP port is not a part of the HostName in the Application attributes.
- In the Domino server, select **Server => Full Access Administrators** should have the name of the user which is being used to open a session with the server.
- Domino Server should be reachable from the IdentityIQ host computer.
- The IQService is a native Windows service that enables IdentityIQ to participate in a Windows environment and access information only available through Windows APIs.
IQService must be installed before performing the following operations:

- Sync ID file
- Upload ID file to vault
- Get ID file from vault
- Reset password of an ID file stored in an ID Vault
- ID File password change through self-service
- HTTP (Internet) Password change through self-service. Helpdesk HTTP (Internet) Password change does not require the IQService.

Ensure that the following pre-requisites are installed on a 32-bit Windows Operating system where IQService needs to be deployed:

- Lotus Notes Client
- Microsoft .NET Framework Version 2.0 or later
- [Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package](#)
- The PATH environment variable should contain the Notes data folder. For example, **C:\Program Files\IBM\Notes**

Administrator permissions

The Administrator user should have Manager Access to the following databases on Domino Server:

- Public Address Book (PAB) Database (default name is **names.nsf**)
- Administration Requests Database (default name is **admin4.nsf**)
- Certification Log Database (default name is **certlog.nsf**)

Configuration parameters

Note: All paths are with respect to the Domino Server computer. For example, ID file path, mail file path and so on. These paths must be accessible from the Domino Server computer.

The following table lists the configuration parameters of Lotus Domino Connector:

| Parameters | Description |
|---------------------------|--|
| IQService Host | Host Name of the computer on which IQService is installed. |
| IQService Port | IQService port number. |
| Admin ID File Path | Administrator ID file path required by IQService. |
| Host Name | Fully qualified host name of the Domino Server. The DIIOP port should be a part of HostName if the HTTP task is not mentioned in the Server Tasks (refer Pre-requisites section). In this case, the HostName should be fullyQualifiedHostName:DIIOPPortNumber. For example, sailpoint.server.com:63148 |
| Admin Name | Name of the Database Administrator which must be in the format Administrator/CertifierName. |
| Admin Password | Password for administrator account. |
| Database Name | Name of the database to be managed. For example, names.nsf |
| Server Name | Name of the server to be managed. For example, Lotus/IBM |
| Search formula - Accounts | Search formula to be used during Account Aggregation. |
| Search formula - Groups | Search formula to be used during Group Aggregation. |
| Indexed database | Specifies if the database is indexed or not. <ul style="list-style-type: none"> ■ Y - The database is indexed ■ N - The database is not indexed <p>Note: A maximum of 5,000 documents will be returned by default. The FT_MAX_SEARCH_RESULTS variable in Notes.ini file overrides this limit for indexed databases or databases that are not indexed but that are running an agent on the client. For a database that is not indexed and is running in an agent on the server, set the TEMP_INDEX_MAX_DOC variable in the Notes.ini file. The absolute maximum value is 2147483647.</p> |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|----------------------|--------------------------------------|
| NOTEID | NOTEID of the user document. |
| Type | The type of the document. |
| Owner | The owner of the document. |
| MailSystem | The type of mail system. |
| InternetAddress | Mail internet address. |
| JobTitle | Job title of the user. |
| CompanyName | Company name of the user. |
| Department | Department of the user. |
| EmployeeID | EmployeeID of the user. |
| Location | Location of the user. |
| Manager | Manager of the user. |
| OfficePhoneNumber | Office phone number of the user. |
| OfficeFAXPhoneNumber | Office fax phone number of the user. |
| CellPhoneNumber | Cell phone number of the user |
| PhoneNumber_6 | Phone number_6 of the user. |
| Assistant | Assistant of the user. |
| OfficeStreetAddress | Office street address of the user. |
| OfficeCity | Office city of the user. |
| OfficeState | Office state of the user. |
| OfficeZIP | Office ZIP of the user. |
| OfficeCountry | Office country of the user. |
| OfficeNumber | Office number of the user. |
| StreetAddress | Street address of the user. |
| City | City of the user. |
| State | State of the user. |
| Zip | Zip/Postal code of the user. |
| Country | Country of the user. |

| Attributes | Description |
|------------------------|--|
| PhoneNumber | Phone number of the user. |
| HomeFAXPhoneNumber | Home fax phone number of the user. |
| Spouse | Spouse of the user. |
| Children | Children of the user. |
| PersonalID | PersonalID of the user. |
| Comment | Office number of the user. |
| WebSite | Address of the user Web Page. |
| PhotoURL | Photo URL of the user. |
| LocalAdmin | Local Admin of the user. |
| CheckPassword | Check password of the user. |
| PasswordChangeInterval | Password change interval of the user. |
| PasswordGracePeriod | Password grace period of the user. |
| PasswordDigest | Password digest of the user. |
| Policy | Policy of the user. |
| Profiles | Profiles of the user. |
| ClientType | Type of the client. |
| PostalAddress | Postal address of the user. |
| HomePostalAddress | Home postal address of the user. |
| Street | Street of the user. |
| BusinessCategory | Business category of the user. |
| CarLicense | Car license of the user. |
| DepartmentNumber | Department number of the user. |
| EmployeeNumber | Employee number of the user. |
| EmployeeType | Employee type of the user. |
| FirstName | First name of the user. |
| MiddleInitial | Middle name initial of the user. |
| LastName | Last name of the user. |
| FullName | Full name of the user. |
| ShortName | Short name of the user. |
| MailDomain | Mail domain of the user. |
| MailServer | Mail server of the user. |
| MailFile | Mail file of the user. |
| PasswordChangeDate | Password change date of the user. |
| HTTPPasswordChangeDate | HTTP password change date of the user. |

Provisioning policy attributes

| Attributes | Description |
|----------------|--|
| SametimeServer | Home sametime server of the user. |
| \$UpdatedBy | Name of the user who last updated the user document. |
| Groups | A list of groups of which the user is a member of. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|-----------------|---|
| GroupType | Type of the group. |
| ListDescription | Description of the group. |
| MailDomain | Mail domain of the group. |
| InternetAddress | Internet address of the group. |
| Comments | Comments about the group. |
| ListOwner | Owner of the group. |
| LocalAdmin | Local admin of the group. |
| ListName | Name of the group. |
| \$UpdatedBy | Name of the user who last updated the group document. |

Provisioning policy attributes

This section lists the different policy attributes of Lotus Domino Connector.

Note: In this section all the attributes marked with the * sign indicate that the attributes are mandatory.

Note: All paths are with respect to the Domino Server computer. For example, ID file path, mail file path and so on. These paths must be accessible from the Domino Server computer.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

| Attributes | Description |
|--------------------|--|
| ServerName* | The name of the server on which the account should be created. |
| CertifierIDfile* | The ID file path of the certifier. For example, c:\id\cert.id |
| CertifierPassword* | The password of the certifier ID file. |
| FirstName | First name of the user. |
| MiddleInitial | Middle name initial of the user. |
| LastName* | Last name of the user. |

| Attributes | Description |
|-----------------------|--|
| UserFullName* | Full name of the user which will be added to the members of the group to which it is to be connected. For example, FirstName LastName/CertifierName. |
| IDFilePath* | ID file path of the user. For example, c:\id\user.id |
| UserIDFilePassword* | ID file password of the user. |
| IDType* | Type of the ID file. Following are the permissible values for the keyword: <ul style="list-style-type: none"> ■ FLAT ■ HIERARCHICAL ■ CERTIFIER |
| MinimumPasswordLength | Minimum length of the ID file password. |
| IDFileIsNorthAmerican | Indicates whether the id file is North American or not. Following are the permissible values for the keyword: <ul style="list-style-type: none"> ■ Y: indicates that the ID file is North American ■ N: indicates that the ID file is not North American |
| StoreIDInAddressbook | Indicates whether the ID file should be stored in the address book or not. Following are the permissible values for the keyword: <ul style="list-style-type: none"> ■ Y: indicates that the ID file should be stored in the address book ■ N: indicates that the ID file should not be stored in the address book. |
| MailServer | Server on which the mail file should be created. |
| MailSystem | Specifies the type of the mail system. Following are the permissible values for the keyword: <ul style="list-style-type: none"> ■ NOTES ■ POP ■ IMAP ■ INOTES ■ INTERNET ■ OTHER ■ NONE |
| MailInternetAddress | Internet address for the mail. |
| MailTemplateName | Name of the mail template. |
| MailForwardingAddress | Forwarding address for the mail. |
| MailFileName | Name of the mail file. For example, mail/mailfilename.nsf |
| MailReplicaServer | The names of the servers on which the mail file replicas should be created. Applies only to clustered servers. Should be multi-valued. |
| CreateMailDatabase | Indicates whether the mail database should be created or not. Following are the permissible values for the keyword: <ul style="list-style-type: none"> ■ Y: indicates that a mail database should be created for the user ■ N: indicates that a mail database should not be created for the user; it will be created during setup. |

Provisioning policy attributes

| Attributes | Description |
|------------------------|---|
| StoreIDInMailFile | Indicates whether the ID should be stored in mail file or not. Following are the permissible values for the keyword: <ul style="list-style-type: none">■ Y: indicates that the ID file should be stored in mail file.■ N: indicates that the ID file should not be stored in mail file. |
| SynchInternetPassword | Indicates whether the ID password and internet password should be in synchronization. Following are the permissible values for the keyword: <ul style="list-style-type: none">■ Y: indicates that the ID file password and internet password should be in synchronization■ N: indicates that the ID file password and internet password should not be in synchronization |
| ExpirationPeriod | The expiration period in years. For example, if 20 is specified and the current year is 2013, the expiration period will be 2033. |
| RegistrationLog | No logging occurs if this parameter is null. If this parameter has a value other than null, logging goes to the certlog.nsf file in the Domino data directory on the registration server. |
| EnforceUniqueShortName | Indicates whether a unique short name should be used. Following are the permissible values for the keyword: <ul style="list-style-type: none">■ Y: indicates that the short name should be unique■ N: indicates that the short name may or may not be unique |
| PolicyName | Name of the explicit policy. |
| RoamingUser | Indicates whether a user is roaming or not. Following are the permissible values for the keyword: <ul style="list-style-type: none">■ Y: indicates that the user is roaming■ N: indicates that the user is not roaming |

Note: All attributes should be of type 'String'.

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|------------|----------------------------------|
| ListName | Name of the group to be created. |

Note: Only the name of the group is required at the time of group creation. Even if the other attributes are specified they will not be set. After creation, the group will be of type 'Multi-Purpose'.

Update policies

The following table lists the attributes for different update policies:

| Attributes | Description |
|------------|---------------|
| | Rename a user |

| Attributes | Description |
|--|---|
| AC_Operation* | The value of this attribute should be Rename User . |
| AC_certifierFilePath* | The location of the Certifier ID file of the user. For example, c:\id\cert.id |
| AC_certifierPassword* | The Certifier ID file password. |
| AC_lastName | New last name of the user. |
| AC_middleInitial | New middle name initial of the user |
| AC_firstName | New first name of the user. |
| AC_orgUnit | New organizational unit of the user. |
| AC_altOrgUnit | New alternate organizational unit of the user. |
| AC_altLanguage | New alternate language of the user. |
| AC_renameNotesUser* | If you want to rename Notes User or not. Values: True or False. |
| Recertify a user | |
| AC_Operation* | The value of this attribute should be Recertify User . |
| AC_certifierFilePath* | The location of the Certifier ID file of the user. For example, c:\id\cert.id |
| AC_certifierPassword* | The Certifier ID file password. |
| Move a user | |
| AC_Operation* | The value of this attribute should be Move User . |
| AC_currentCertifierFilePath* | The location of the Certifier ID file of the user. For example, c:\id\cert.id |
| AC_currentCertifierPassword* | The Certifier ID file password. |
| AC_targetCertifierFilePath* | The location of the Certifier ID file of the user. For example, c:\id\target.id |
| AC_targetCertifierPassword* | The Certifier ID file password. |
| AC_targetCertifierName* | The name of the target certifier. |
| Change/Reset password of a user | |
| HTTP_PASSWORD_CHANGE | Should be set to Yes to change the HTTP (Internet) Password of a user. Values: Yes and No (Default). |
| IDFilePath | The location where the user ID file is stored. For example, c:\id\user.id should be provided to change the ID file password of a user. |
| RESET_PASSWORD | Should be set to Yes to reset the password of an ID file stored in the vault. Values: Yes and No (Default). |
| Sync ID File | |
| Operation* | The value of this attribute should be Sync ID File . |
| IDFilePath* | The location of the User ID file of the user. For example, c:\id\user.id |
| IDFilePassword* | The User ID file password. |

Install and register the IQService

| Attributes | Description |
|-----------------------|---|
| Get ID File | |
| Operation* | The value of this attribute should be Get ID File . |
| IDFilePath* | The location where the User ID file should be stored. For example, c:\id\user.id |
| IDFilePassword* | The User ID file password. |
| Upload ID File | |
| Operation* | The value of this attribute should be Upload ID File . |
| IDFilePath* | The location where the User ID file is stored. For example, c:\id\user.id |
| IDFilePassword* | The User ID file password. |

Note: No default value needs to be assigned for optional attributes if those need not to be set.

Install and register the IQService

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the **IQService.zip** archive from the **identityIQHome\WEB-INF\bin\win** directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.
IQService.exe -i
4. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information

Additional information

This section describes the additional information related to the Lotus Domino Connector.

ID Vault functionalities

ID Vault is a feature introduced by IBM in Domino version 8.5. The following functionalities are a part of ID Vault which are supported through IQService:

- **Reset Password:** Allows the Help Desk Personnel to reset the password of the ID file for a vaulted user. This requires application Administrator to have password reset authority.
- **Extract ID file from vault:** A vault administrator assigned to the Auditor role in the vault database ACL can extract an ID from a vault to gain access to a user's encrypted data. A copy of the ID remains in the vault after extraction.
- **Upload ID file to vault:** Upload an ID file that has not yet been uploaded to the vault.
- **Sync ID file:** Synchronizes the local ID file with the copy in ID vault.

Check the provisioning policy for each of the above transactions.

Password management

Administrator password reset (Change password for others):

- HTTP Password
- Password of the vaulted ID file (Reset password)

Self-service password change:

- HTTP Password
- ID File Password
- Password of the ID file which is vaulted (Reset password)

Troubleshooting

1 - Could not get IOR from Domino Server

Resolution: Perform the following:

1. Check if the Domino Server is accessible from the IdentityIQ computer using the Fully Qualified Internet Host Name. The ping should be successful using the Fully Qualified Internet Host Name of the Domino Server.
2. Check if DIIOP is present in the ServerTasks of **notes.ini** file.
3. If HTTP is not added to **notes.ini** file ServerTasks, the HostName in the Application Parameter should include the port number of the DIIOP Server in the following format:
`fullyQualifiedInternetHostName:DIIOPPortNumber`
 For example, **LOTUS-AME.SAILPOINT.COM:63148**
4. Check if **NCSO.jar** file is present in the CLASSPATH environment variable.

2 - Could not open the ID file

Resolution: Perform the following:

1. All paths in the connector are with respect to the Domino Server. Verify if the ID file path you have provided is accessible from the Domino Server computer.

Troubleshooting

2. The ID files will be read from and created on a path with respect to the Domino Server.

3 - Add Account gives Object does not exist exception

Resolution: Perform the following:

1. If the name of the user you created is Derek Stevens and the name of the certifier under which the user was created is /USA then the following attributes should be populated:
 - FirstName: Derek
 - LastName: Stevens
 - FullName: Derek Stevens/USA

Add account searches a user based on the FullName of the user, hence it is important that it is provided correctly.

4 - IQService - Unable to load DLL 'SPLotusNotesWrapper.dll': The specified module could not be found or the IQService stops responding.

Resolution: Perform the following:

1. Verify if the PATH system variable contains the Notes data folder. For example, c:\Program Files\IBM\Notes should be present in the PATH system variable.
2. Verify if you have restarted the computer after modifying the PATH system variable.
3. Close the Notes Administrator/Client and restart the IQService.
4. Copy the IQService installation files in the Notes folder of IBM Lotus Notes Client. For example: **C:\Program Files\IBM\Notes**

Chapter 18: SailPoint IdentityIQ Microsoft SQL Server

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 157 |
| Supported features | 157 |
| Supported Managed Systems | 158 |
| Pre-requisites | 158 |
| Administrator permissions | 158 |
| Configuration parameters | 158 |
| Schema attributes | 159 |
| Account attributes | 159 |
| Group attributes | 160 |
| Provisioning Policy attributes | 160 |
| Additional information | 160 |
| Delete login | 161 |
| Direct permission | 161 |
| Identity and Entitlement representation | 161 |

Overview

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).

SailPoint IdentityIQ Microsoft SQL Server Connector manages the following entities on Microsoft SQL Server:

- User
 - Login User
 - Database User
- Role
 - Application Role
 - Database Role

Supported features

SailPoint IdentityIQ Microsoft SQL Server Connector provides support for the following features:

- Account/Group Aggregation
- Create/Update/Delete/Refresh Account
- Create/Delete Group
- Enable/ Disable Account
- Set Password
- Request/Remove Entitlement
- Direct Permissions

For more information, see [“Direct permission” on page 161](#).

Supported Managed Systems

Following versions of Microsoft SQL Server is supported by the SailPoint IdentityIQ Microsoft SQL Server Connector:

- Microsoft SQL Server 2005 with SP1
- Microsoft SQL Server 2005 with SP2
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012

Pre-requisites

The **Sqljdbc.jar** file is required for connecting to the Microsoft SQL Server using JDBC.

Administrator permissions

The Administrative Account login must have the following minimum privileges:

- sysadmin (server fixed role)
- securityadmin (server fixed role)
- public (database fixed roles) in each database

Configuration parameters

The following table lists the configuration parameters of SailPoint IdentityMicrosoft SQL Server Connector:

| Parameters | Description |
|--------------------|---|
| URL* | A valid URL of Microsoft SQL Server with the following format: jdbc:sqlserver://[serverName\[instanceName\]:portNumber] <ul style="list-style-type: none">■ jdbc:sqlserver://: (<i>Required</i>) is known as the sub-protocol and is constant■ serverName: is the address of the server to connect to. This could be a DNS, IP address, localhost, or 127.0.0.1 for the local computer.■ instanceName: is the instance to connect to <i>serverName</i>.■ portNumber: is the port to connect to <i>serverName</i>. The default is 1433. |
| User* | Administrative Account to connect to Microsoft SQL Server. |
| Password* | Administrative Account password. |
| Driver* | The name of the Driver class supported by JDBC com.microsoft.sqlserver.jdbc.SQLServerDriver |
| Included Databases | List of comma separated databases names to be included in the aggregation operation. |

| Parameters | Description |
|--------------------|---|
| Excluded Databases | <p>List of comma separated databases name to be excluded in the aggregation operation.</p> <p>Note: If the Include Database paramter is populated, the exclude database paramter would be ignored.</p> |

Note: All the parameters in the above table marked with the * sign are mandatory parameters.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attribute name | Description |
|---------------------|--|
| native_identity | Native identity represented by default as <i>loginName@serverName</i> or <i>loginName@databaseName</i> . |
| database_name | Database name of the login. |
| database_id | Database Id. |
| server_login | Server login associated to account. |
| name | Account name. |
| principal_id | ID of database principal. |
| type | Type of the login. |
| type_desc | Description type of the login. |
| default_schema_name | Name to be used when SQL name does not specify schema. |
| create_date | Creation date of the login. |
| modify_date | Last modification date of the login. |
| sid | SID of the login. |
| server_name | Server name. |
| is_fixed_role | If the value is 1, then this row represents an entry for one of the fixed database roles. |
| owning_principal_id | ID of the principal that owns this database principal. |
| roles | Roles assigned to the login. |
| DBUser | Database users which are associated to the login. |

Group attributes

The following table lists the group attributes:

| Attribute name | Description |
|-----------------|--|
| native_identity | Native identity represented by default as <i>groupName@serverName</i> or <i>groupName@databaseName</i> . |
| name | Group name. |
| database_name | Database name in which group exists. |
| database_id | Database ID in which group exists. |
| principal_id | Id of database principal. |
| roles | Roles assigned to the group. |
| server_name | Server name of the group. |
| type_desc | Description type of the group. |

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Account and Create Group:

| Attribute name | Description |
|-----------------------|--|
| Create Account | |
| Login name* | Server login name. |
| password | Password of server login name. |
| Account Type* | Type of the server login name. Note: Either one of the attributes (Windows Login or SQL Login) is Yes . |
| User Mapping | User mapping format (Username@datasenname). |
| Create Group | |
| Group name* | Group name. |
| Group Type* | Type of the Group. Note: Either one of the attributes (Application Role or Database Role) is Yes . |
| Password | Password for Application type role. |

Additional information

This section describes the additional information related to the Microsoft SQL Server Connector.

Delete login

By default the Connector would not delete the login users which have associated database users. In order to override this behavior, the **DeleteLoginBYdefault** configuration parameter must be set to **Y**.

Direct permission

Following targets are supported:

- DATABASE
- SERVER
- PROCEDURE

For example, **GRANT CONNECT on DATABASE_**DatabaseName
GRANT CONNECT on SERVER_ServerName

{ [GRANT/REVOKE] [SELECT/ INSERT/ DELETE/UPDATE] ON Table TO Account}

Note: **DATABASE_** is appended before DatabaseName. Similar appending is done for other objects such as **SERVER_** for ServerName, **PROCEDURE_** for Procedures.

Identity and Entitlement representation

This section describes the Identity and Entitlement representation for SailPoint IdentityIQ Microsoft SQL Server Connector.

Identity representation

Account: The Account in Microsoft SQL Server Connector is represented as follows:

<Account name>@<Container name>

- For Database login it is represented as **<Account name>@<Database name>**
 For example, username@master
- For Server Login it is represented as **<Account name>@<Server name>**
 For example, loginname@MSSERVER

Entitlement representation

Groups: The Groups in Microsoft SQL Server Connector are represented as follows:

<Group name>@<Container name>

- For Database Role and Application Role it is represented as **<Group name>@<Database name>**
 For example, userDatabaseRole@master
- For Server roles it is represented as **<Group name>@<Server name>**
 For example, userGroup@MSSERVER

Additional information

Chapter 19: SailPoint IdentityIQ Oracle Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 163 |
| Supported features | 163 |
| Supported Managed Systems | 163 |
| Pre-requisites | 164 |
| Administrator permissions | 164 |
| Configuration parameters | 165 |
| Schema attributes | 165 |
| Account attributes | 165 |
| Group attributes | 166 |
| Provisioning policy attributes | 166 |
| Troubleshooting | 167 |

Overview

The Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system (ORDBMS).

SailPoint IdentityIQ Oracle Server Connector is a connector to Oracle database server that allows full user administration with provisioning and password management capabilities of Oracle server. Oracle Server Connector manages the following entities of Oracle server:

- Account
- Role

Supported features

SailPoint IdentityIQ Oracle Connector provides support for the following features:

- Account/Group Aggregation
- Create/Update/Delete/Refresh Account
- Request/Remove Entitlement
- Enable/Disable Account
- Set Password
- Pass through Authentication
- Create/Update/Delete Group
- Direct Permissions: Target is Table

Supported Managed Systems

- Oracle Server version 11g
- Oracle Server version 10g

Pre-requisites

Type 4 thin driver is required for connecting to the Oracle server using JDBC.

By default IdentityIQ is bundling **ojdbc14.jar** file (which works properly with Oracle 10i families). But if the backend Oracle Server is of version greater than 10i, the connector does not function properly with **ojdbc14.jar** file. For more information, see the [“Troubleshooting” on page 167](#) section.

Administrator permissions

The Oracle administrator must have all the permissions mentioned below for performing the provisioning operations.

Permissions required for Get operations

Grant SELECT privilege on the following tables:

- dba_users
- dba_roles
- dba_sys_privs
- dba_role_privs
- dba_tab_privs
- dba_col_privs
- dba_sys_privs
- system_privilege_map
- v\$version

Permissions required for Set operations

In addition to the Get operations permission, following permissions are required for Set operations.

Grant following privileges:

- CREATE SESSION
- CONNECT
- SELECT ANY TABLE
- CREATE USER with admin option
- ALTER USER
- DROP USER
- CREATE ROLE with admin option
- ALTER ANY ROLE
- DROP ANY ROLE
- GRANT ANY ROLE
- GRANT ANY PRIVILEGE
- CREATE PROFILE
- ALTER PROFILE

- DROP PROFILE
- CREATE SESSION
- EXECUTE on SA_USER_ADMIN (if it exists)
- EXECUTE on dbms_resource_manager_privs
- EXECUTE on SA_SYSDBA (if it exists)

Configuration parameters

The following table lists the configuration parameters of SailPoint IdentityIQ Oracle Connector:

| Parameters | Description |
|--------------|---|
| url* | <p>The url to connect to the database. The format is jdbc:oracle:thin:@serverName:PortNumber:SID</p> <p>For example jdbc:oracle:thin:@172.16.21.31:1521:ORCL url consist of</p> <ul style="list-style-type: none"> ■ jdbc:oracle:thin:@: This is common part which states that the connection is made using thin driver. ■ 172.16.21.31: server Name or IP of the oracle server ■ 1521: The port number of the oracle server. This port number should be known by the oracle server administrator. ■ ORCL: The SID of the oracle server. This port number should be known by the oracle server administrator. |
| user* | Name of the administrative account which has all the privileges to perform the CRUD (Create, Read, Update, and Delete) operations. The default administrator of Oracle Server is System . |
| password* | The password of Administrative account. |
| driverClass* | Name of the type4 driver to use when making connection with oracle server. By default this connector uses oracle.jdbc.driver.OracleDriver |

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Provisioning policy attributes

| Attribute name | Description |
|---------------------|---|
| USERNAME | User name. |
| USER_ID | User ID. |
| ACCOUNT_STATUS | Account status. |
| DEFAULT_TABLESPACE | Default tablespace. |
| ROLES | Roles assigned to the user. |
| PROFILES | Profiles assigned to the user. |
| TEMP_TABLESPACE | Temporary tablespace. |
| SYSTEM_PRIVILEGES | System privileges assigned to the user. |
| AUTHENTICATION_TYPE | Authentication type. |

Group attributes

The following table lists the group attributes:

| Attribute name | Description |
|---------------------|---|
| ROLE | Role name. |
| AUTHENTICATION_TYPE | Authentication type. |
| SYSTEM_PRIVILEGES | System privileges assigned to the role. |
| ROLES | Roles assigned to the role. |
| PASSWORD_REQUIRED | Password is required or not. |

Provisioning policy attributes

The following table lists the provisioning policy attributes for Create Account and Create Group:

| Attribute name | Description |
|------------------------------------|------------------------|
| Create Account | |
| Username* | Username |
| password* | Password of the user. |
| DEFAULT_TABLESPACE | Default tablespace. |
| TEMP_TABLESPACE | Temporary tablespace. |
| PROFILE | Profile |
| AUTHENTICATION_METHOD | Authentication method. |
| Create Group | |
| GROUP_NAME (Role or Profile Name)* | Role or profile name. |

| Attribute name | Description |
|-------------------|---|
| Roles | Role to assign for new role. |
| System Privileges | System privileges to assign for new role. |

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Troubleshooting

1 - If the version of Oracle Server is greater than 10i, the connector does not work properly with ojdbc14.jar file.

For example, for Oracle version 11g the ojdbc jar file which is required for proper functioning of the connectors is **ojdbc6.jar** file.

Resolution: Perform the following to obtain the required version of the ojdbc jar file:

1. Download the suitable ojdbc jar file for oracle from the <http://www.oracle.com> site to a temporary directory.
2. Turn off the webserver.
3. Rename the already bundled **ojdbc14.jar** file to **ojdbc14.jar_old** in the **..\identityiq\WEB-INF\lib** directory.
4. Copy the latest downloaded jar file in step 1 from the temporary directory to the **..\identityiq\WEB-INF\lib** directory.
5. Restart the webserver.

Chapter 20: SailPoint IdentityIQ Sybase Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 169 |
| Supported features | 169 |
| Supported Managed Systems | 170 |
| Pre-requisites | 170 |
| Administrator permissions | 170 |
| Configuration parameters | 170 |
| Schema attributes | 171 |
| Account attributes | 171 |
| Group attributes | 172 |
| Provisioning policy attributes | 172 |
| Additional information | 172 |
| Delete login | 173 |
| Direct Permissions | 173 |
| Identity and Entitlement representation | 173 |
| Troubleshooting | 174 |

Overview

Sybase Adaptive Server Enterprise (ASE) is widely used database Server, mainly used to store data for different business modules like Sales, Production, Human Resource, Finance and Accounting. It requires the user to authenticate in order to connect to database to manipulate business data. It controls the users/roles logging in to Sybase ASE Managed System and performs other activities like processing transactions, writing logs, updating database files and so on.

A group is a means of organizing users, where as a role is usually a means of organizing rights. User roles are aggregated as Account Groups as it is widely used by the customers.

SailPoint IdentityIQ Sybase Adaptive Server Enterprise Connector manages the following entities on Sybase Adaptive Server Enterprise:

- Login User
- Database User
- Roles

Supported features

SailPoint IdentityIQ Sybase Adaptive Server Enterprise Connector provides support for the following features:

- Account/Group Aggregation
- Refresh/Create/Update/Delete/Enable/Disable Accounts
- Create/Update/Delete Group
- Set password
- Request/Remove Entitlement
- Direct Permission

For more information, see [“Direct Permissions” on page 173](#).

Supported Managed Systems

Following versions of Sybase ASE are supported by the SailPoint IdentityIQ Sybase ASE Connector:

- Sybase ASE 15.0
- Sybase ASE 15.0.1
- Sybase ASE 15.0.2
- Sybase ASE 15.0.3
- Sybase ASE 15.5
- Sybase ASE 15.7

Pre-requisites

Sybase JDBC Driver is required for proper functioning of SailPoint IdentityIQ Sybase ASE Connector. This JDBC driver must be copied in the `..\identityiq\WEB-INF\lib` directory. For example, the `jconn4.jar` jar can be downloaded from the <http://www.sybase.com>.

Administrator permissions

Following are the minimum Administrative Account permissions required to be granted:

- SSO_ROLE
- SA_ROLE

Configuration parameters

The following table lists the configuration parameters of SailPoint IdentityIQ Sybase ASE Connector:

| Parameters | Description |
|--------------------|--|
| url* | A valid URL of Sybase ASE Connector which directly interacts with the managed system. In case of <code>jconn2.jar</code> , use the following url: <code>jdbc:sybase:Tds:<host>[:<port>]</code> For example, <code>jdbc:sybase:Tds:ACHAUDHARI:5000</code> |
| user* | Administrative Account to connect to Sybase ASE. |
| password* | Administrative Account Password. |
| driverClass* | The name of the Driver class supported by JDBC Type 4. For example, In case of <code>jconn2.jar</code> , use the following driverClass: <code>com.sybase.jdbc2.jdbc.SybDataSource</code> |
| Included Databases | List of comma separated database names to be included in the aggregation operation. |

| Parameters | Description |
|--------------------|---|
| Excluded Databases | <p>List of comma separated database names to be excluded in the aggregation operation.</p> <p>Note: If the Include Database paramter is populated, the Exclude Database paramter would be ignored.</p> |

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attribute name | Description |
|---------------------|---|
| name | Login user name. |
| DB User Name Detail | DB User associated with Login User. |
| server_user_id | Server User ID. |
| user_id | User ID. |
| default_database | Default database. For example: master |
| default_language | Default language. |
| full_name | Full name of login user. |
| create_date | Date on which login user is created. |
| lock_date | Date on which login user got locked. |
| password_chg_date | Date on which password got changed. |
| last_login_date | Last login date of the user. |
| user_type | Type of User: Login or Database user. |
| native_identity | Native identity is an attribute which act like primary key. |
| status | Status of login user: enable/disable |
| roles | Roles associated with login user. |
| groupname | Database user's group name. |

Group attributes

The following table lists the group attributes:

| Attribute name | Description |
|-------------------|--|
| server_role_id | ID of the server Role. |
| native_identity | Native identity is an attribute which act like primary key during aggregation. |
| name | Name of the Role. |
| password_chg_date | Date on which password got changed. |
| member_roles | Roles which are present under the hierarchy of the main role. |

Provisioning policy attributes

This section lists the single provisioning policy attributes of SailPoint IdentityIQ Sybase ASE Connector that allows to select the type of user, login, or group.

| Attribute name | Mandatory |
|--------------------------------------|---|
| Creating Group (User Role) | |
| Role name* | Name of the role created. |
| Parent Container | Parent container in which role should be created. |
| Creating User (Login User) | |
| Name* | Name of the Login User. |
| password* | Password for LoginUser. |
| Default database | Default database for Login User. |
| Default language | Default language. |
| Full name | Full name of the Login User. |
| Creating User (Database User) | |
| DB User Name Detail | Database user with details. For example, The format of DB User Name Detail attribute must be as <DB User Name>@<Server Name>#<Group Name>. For example, SailPointDBUser@master#public . |

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Additional information

This section describes the additional information related to the Sybase Connector.

Delete login

By default the Connector would not delete the login users which have associated database users. In order to override this behavior, the **DeleteLoginBYdefault** configuration parameter must be set to **true**.

Direct Permissions

Permissions on the following targets:

- TABLE
- VIEW
- COLUMNS
- STORED PROCEDURES

SailPoint IdentityIQ Sybase ASE Connector provisions multiple targets by appending the names of the target with its type.

For example,

- if the target is TABLE and the name of table is XXXX.
Then for TABLE it would be **TABLE:XXXX**
- if the target is VIEW and the name of table is XXXX.
Then for TABLE it would be **VIEW:XXXX**

Similar to COLUMNS and STORED PROCEDURES

Identity and Entitlement representation

This section describes the Identity and Entitlement representation for SailPoint IdentityIQ Sybase Adaptive Server Connector.

Identity representation

Account: The Account in Sybase ASE Connector is represented as follows:

<Account name>@<Server name\Database name>

- For Database login it is represented as **<Account name>@<Database name>**
For example, username@master
- For Server Login it is represented as **<Account name>@<Server name>**
For example, loginname@SYBASE

Entitlement representation

Groups: The Groups in Sybase ASE Connector are represented as follows:

<Group name>

- For Application Role it is represented as **<Group name>**

Troubleshooting

1 - Delete Login User and associated DBuser fails with an error

When **DeleteLoginByDefault** is set as **true**, the user is trying to delete a login user (for example, `<xyz@ServerName>`) and the associated dbuser (for example, `<abc1@master>`) from IdentityIQ Console hence the following error messages are displayed:

User `<abc1@master>` not found in database master

User `<abc2@model>` not found in database master

This scenario occurs as the Login User has been deleted along with the associated Database users.

Chapter 21: SailPoint IdentityIQ Windows Local Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|-----|
| Overview | 175 |
| Supported features | 175 |
| Supported Managed Systems | 176 |
| Pre-requisites | 176 |
| Administrator permissions | 176 |
| Configuration parameters | 176 |
| Schema attributes | 177 |
| Account attributes | 177 |
| Group attributes | 178 |
| Provisioning Policy attributes | 178 |
| Install and register IQService | 179 |
| Additional information | 179 |
| Unstructured Target Collector | 180 |
| Troubleshooting | 181 |

Overview

SailPoint IdentityIQ Windows Local Connector manages User Accounts and Groups on Windows Operating System based computers through IQService. IQService uses WinNT ADSI service provider to connect to local users/groups for all versions of windows.

Supported features

SailPoint IdentityIQ Windows Local Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Account refresh
- Create\Update\Delete account
- Create\Update\Delete group
- Add\Remove Entitlement
- Enable\Disable\Unlock account
- Password change\reset
- Target aggregation

For more information, see [“Unstructured Target Collector”](#) on page 180.

- Revoke target permissions

Supported Managed Systems

Following versions of Microsoft Windows are supported by the SailPoint IdentityIQ Windows Local Connector:

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Microsoft Windows Vista
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008 R2
- Microsoft Windows 7
- Microsoft Windows 8
- Microsoft Windows Server 2012

Pre-requisites

- Microsoft .NET Framework Version 2.0 or later must be installed on the system running IQservice.
- Remote registry service must be started on the target system.
- Allow Exception for **File and Printer Sharing** in windows firewall.
- To **Turn off User Account Control** for Microsoft Windows Vista or later, perform the following steps:
 - a. For Microsoft Windows Vista and Microsoft Windows Server 2008, open the **Control panel ==> User Accounts ==> Turn User Account Control on or off**
 - b. For Microsoft Windows 7 onwards, open the **Control panel ==> User Accounts ==> User Accounts ==> Change User Account Control settings**

Administrator permissions

User should be a member of **Administrators** group of Windows host computer which is to be managed.

Configuration parameters

The following table lists the configuration parameters of SailPoint IdentityIQ Windows Local Connector:

| Parameters | Description |
|-------------------------------|--|
| IQService Host* | Host name or IP address where IQservice is installed. |
| IQservice port* | The TCP/IP port where the IQService is listening for requests (Default: 5050). |
| UserName* | User name of the account with administrator rights on the managed system (Syntax: <i>computerName\userName</i> or <i>userName</i>). For domain users it will be: <i>domainName\userName</i> |
| Password* | Password of user account mentioned in UserName field. |
| Server* | Host name or IP address of windows computer which is to be managed. |
| disableQualifyingLocalObjects | Flag to indicate whether aggregated objects must not be prefixed with server name. (Defaults to false. If set to true then aggregated object will not be prefixed with server name). |

| Parameters | Description |
|------------|--|
| pageSize | Number of objects to fetch in a single request. Defaults to 1000 |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|----------------------|--|
| AutoUnlockInterval | Time interval for auto unlocking of locked user account. |
| Disabled | Flag to indicate if the user is disabled. |
| Description | User's description. |
| DirectoryPath | Fully qualified directory path WinNt://... |
| FullName | User's fullname. |
| groups | List of groups assigned to a user. |
| HomeDirectory | Location of the user's home directory. |
| Lockedout | Flag to indicate a user is locked out. |
| MaxStorage | The maximum amount of disk space the user can use. |
| MinPasswordLength | Minimum length of the user's password. |
| Name | Name of the account unqualified SAMAccountName. |
| objectSid | Windows SID. |
| PasswordAge | Time duration of the password in use. This property indicates the number of seconds that have elapsed since the password was last changed. |
| PasswordExpired | Indicates if the password is expired. |
| PasswordNotRequired | Flag to indicate if the user requires a password. |
| PasswordUnchangeable | Flag to indicate if the user password can be changed. |
| Profile | User's profile. |
| PrimaryGroupID | ID of the user's primary group. |
| sAMAccountName | Fully qualified version of the sAMAccountName. |
| UserFlags | User Flag defined in ADS_USER_FLAG_ENUM. |
| BadPasswordAttempts | Number of consecutive Bad Password Attempts made last time. |
| LoginScript | File path of Login script file. |
| HomeDirDrive | Home Directory Drive of the user. |

Provisioning Policy attributes

| Attributes | Description |
|----------------------|--|
| PasswordNeverExpires | Flag to indicate if the password never expires. |
| MaxPasswordAge | Indicates the maximum time interval, in seconds, after which the password must be changed. |
| MinPasswordAge | Indicates the minimum time interval, in seconds, before the password can be changed. |
| LastLogin | Date and time when user logged in last time. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|----------------|--|
| Description | User's description. |
| DirectoryPath | Fully qualified directory path WinNt://... |
| GroupMembers | List of groups assigned to a group. |
| GroupType | Windows SID. |
| Members | List of users assigned to a group. |
| objectSid | Windows SID. |
| sAMAccountName | Fully qualified version of the sAMAccountName. |

Provisioning Policy attributes

This section lists the provisioning policy attributes of SailPoint IdentityIQ Windows Local Connector for create Account, create Group, and update Group.

| Attributes | Description |
|---|--|
| For create Account | |
| sAMAccountName* | Name for user account to create. (Syntax: if disableQualifyingLocalObjects attribute in application configuration is unchecked then the format is sAMAccountName = <i>hostName\userName</i> . Otherwise sAMAccountName = <i>userName</i> .) |
| Password* | Password for new user account. |
| Description | Description of new user account. |
| Full Name | Full name of the user account. |
| Disable user account | Flag to create disabled user account. |
| User must change password on next logon | Flag to indicate whether user must change his password on next logon. |

| Attributes | Description |
|-----------------------------|--|
| User cannot change Password | Flag to indicate whether user is allowed to change his password. If the value is false , user can change his password. Otherwise only system administrator can change his password. |
| Password never expires | Flag to indicate that user account password never expires until next password set. |
| For create Group | |
| sAMAccountName* | Name for group to create. (Syntax: if disableQualifyingLocalObjects attribute in application configuration is unchecked then sAMAccountName = <i>hostName\groupName</i> . Otherwise sAMAccountName= <i>groupName</i>). |
| For update Group | |
| Description | Description of the group. |
| GroupType | Type of the group. |
| objectSid | Windows SID of group. |
| DirectoryPath | Fully qualified directory path WinNt://... |

Note: Attributes marked with * sign are the mandatory attributes.

Install and register IQService

To install and register IQService, perform the following:

1. Create a directory in which you want to download the service. For example, **c:\iqservice**.
2. Extract the IQService.zip archive from the **IIQHOME\WEB-INF\bin\win** directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service named IQService.
IQService.exe -i
4. Start the service either from the Services Applet or from the command line by running the following command:
IQService.exe -s

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information

Additional information

This section describes the additional information related to the Windows Local Connector.

Unstructured Target Collector

Windows Local unstructured target collector supports aggregating direct access permissions on resources such as shared files and folders from target system and correlate it with aggregated user accounts and groups using objectSid as the correlation key.

Pre-requisites for target aggregation

IQService needs to be installed on Target Windows computer.

Target aggregation configuration parameters

The following table lists the different target aggregation configuration parameters:

| Attributes | Description |
|--|---|
| IQService configuration parameters | |
| IQService Host* | The host on which the IQService resides. |
| IQService Port* | The TCP/IP port where the IQService is listening for requests. |
| Number of targets per block | Number of targets (files) to include in each block of data returned. |
| File share configuration parameters | |
| Path* | Path of file or directory. You can target a specific file or a directory and its sub-directories containing multiple files from which to extract the required data. If you target a directory, use the Wildcard and Directory Depth fields to narrow the query if possible. |
| Directories Only | Use to instruct the collector to ignore files and just report back directory permission information. Valid only if Path value is directory path. |
| Directory Depth | The sub-directory depth from which to extract data. The Directory Depth field enables you to extend your query up to ten (10) sub-directories below the one specified in the Path field. |
| Wildcard | Use wild cards to target a particular file type or naming scheme. For example, to search only exe, use *.exe or to search only files with names beginning with New_ and New_*.* |
| Administrator* | The administrator that has access to this share so you can collect permissions. This value can be domain\userName, computerName\userName, or userName. |
| Password* | The password associated with the specified administrator. |
| Rule configuration parameters (used to transform and correlate the targets) | |
| Creation Rule | The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ. |
| Correlation Rule* | The rule used to determine how to correlate account information from the application with identity cubes in IdentityIQ. |

Note: Attributes marked with * sign are the mandatory attributes.

Troubleshooting

1 - Error returned from IQService: Unspecified Error

The following error message is displayed for any Windows Local application request:

Error returned from IQService:Unspecified Error

Resolution: Perform the following:

1. Ensure that the Target system is up and accessible from IQservice host.
2. Ensure that the Username and Password provided in application configuration are correct.
3. If the target system is in a workgroup **Guest Only** option for **Sharing and security model for local accounts** in local policy will force all incoming network file sharing connections to authenticate as **Guest**.
To resolve this problem perform the following steps:
 - a. On the Windows Start menu, click **Start ==> Control Panel ==> Administrative Tools ==> Local Security Settings**.
 - b. In the left pane, expand **Local Policies ==> Security** options.
 - c. In the right pane, double-click **Network access: Sharing and security model for local accounts**.
 - d. Select **Classic - local users authenticate as themselves** and click **OK**.
4. If the target system is Windows Server 2003 Service Pack 2 then some Windows updates are missing from the system. Turn on the Windows updates and install the latest updates.
5. Ensure that exception for **File and Printer Sharing** in windows firewall is enabled.
6. If the problem still persists try restarting IQservice.

2 - Error returned from IQService: The network path was not found

When Remote registry service is not started on Windows computer the following error message is displayed:

Error returned from IQService: The network path was not found

Resolution: Ensure that Windows Service named, **Remote Registry Service** is started on the Windows managed system.

3 - Unspecified Error

The following error message is displayed for any Windows Local Connector operation after upgrading to latest version from version 6.0 Patch 5 or below.

Unspecified Error

Resolution: Perform following:

1. Navigate to IdentityIQ debug page.
2. Select **Application** from the object browser.
3. Select and open your application from the list.
4. If a line exists with the following text as the starting text, then delete the line and save the application
`"<entry key="domain" "`

Chapter 22: SailPoint IdentityIQ AIX Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 183 |
| Supported features | 183 |
| Supported Managed Systems | 184 |
| Pre-requisites | 184 |
| Administrator permissions | 184 |
| Configuration parameters | 184 |
| Additional configuration parameters for SSH configuration | 185 |
| Public key authentication configuration. | 185 |
| Schema attributes | 186 |
| Account attributes | 186 |
| Group attributes. | 192 |
| Provisioning policy attributes | 192 |
| Account attributes | 192 |
| Group attributes. | 193 |
| Additional information | 193 |
| Additional information. | 193 |
| Troubleshooting | 194 |

Overview

AIX Connector was developed to manage the accounts and groups on AIX computer.

Supported features

The AIX Connector provides support for the following features:

- Account Aggregation
- Account Group Aggregation
- Create/Update/Delete Account
- Get/Sync Account
- Enable/Disable/Unlock Account
- Change Password
- Create/Update/Delete Account Group
- Add/Delete entitlement
- Reset password
- Target Aggregation
- Revoke Target Permissions
- Password Interceptor

For more information, see [“Unstructured Target Collector”](#) on page 194.

Password Interceptor for AIX provides the mechanism by which a password change initiated from AIX system is captured by the Client and sent to IdentityIQ. For more information, see [“Password Interceptor for UNIX”](#) section in [Appendix A: Password Interceptor](#).

Configuration parameters

Note: Role provisioning is not supported by AIX Connector.

Supported Managed Systems

The AIX connector supports the following versions of the operating system:

- AIX 7.1
- AIX 6.1
- AIX 5.3

Pre-requisites

SSH should be installed on AIX computer.

Administrator permissions

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

Rights to execute the following commands with root privilege:

```
/usr/sbin/luser, /usr/sbin/lsgroup, /usr/bin/chmod, /usr/bin/mkuser,  
/usr/sbin/userdel, /usr/bin/chuser, /usr/bin/chgroup, /usr/bin/mkgroup,  
/usr/sbin/rmggroup, /usr/bin/passwd, /bin/rm, /bin/echo, /usr/bin/find,  
/usr/bin/pwdadm
```

An entry in /etc/sudoers file should look similar to the following:

```
username ALL = (root) PASSWD : /usr/sbin/luser, /usr/sbin/lsgroup,  
/usr/bin/chmod, /usr/bin/mkuser, /usr/sbin/userdel, /usr/bin/chuser,  
/usr/bin/chgroup, /usr/bin/mkgroup, /usr/sbin/rmggroup, /usr/bin/passwd,  
/bin/rm, /bin/echo, /usr/bin/find, /usr/bin/pwdadm
```

Note: All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on AIX computers as they might differ from the values mentioned here.

Note: If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

Configuration parameters

The following table lists the configuration parameters of AIX Connector:

| Parameters | Description |
|------------------|--|
| Unix Server Host | Host Name/IP address of AIX computer. |
| SSH Port | SSH port configured. Default value: 22 |

| Parameters | Description |
|----------------------------|--|
| Not a 'root' user | If User ID specified is not root, check this paramter. |
| User Name | User ID on AIX computer that you want to use for connector operations. |
| User Password | Password of the target system user account that you want to use for connector operations. |
| Private Key File Path | Path to Private Key File. Private/Public key authentication will have precedence over password authentication. |
| Passphrase For Private Key | Passphrase provided for creating Private Key. |

Additional configuration parameters for SSH configuration

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

Note: These additional configuration parameters must be added in the Application/Target Source debug page.

- Following is the default command for setting shell prompt on UNIX computer:
`<entry key="SetPrompt" value="PS1='SAILPOINT>'"/>`
 In the above command, “SetPrompt” is the application/target source attribute and PS1=‘SAILPOINT’ is the value of the application/target source attribute.
 If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:
 For example: For tcsh shell, the entry value would be:
`<entry key="SetPrompt" value="set prompt='SAILPOINT>'"/>`
- For executing the commands, verify that the default shell is present on your system.
 If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:
`<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>`

Public key authentication configuration

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

- Generate Private and Public key's. For more information of the standard steps, see [“5 - Test connection fails for key based authentication with an error.” on page 195.](#)
- Append contents of public key file to `~/.ssh/authorized_keys` as shown below.
`cat <public key file> >> ~/.ssh/authorized_keys`

Schema attributes

3. Copy private key file to a location which is accessible by IdentityIQ server.
4. Provide path of private key file in application configuration.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|------------|--|
| User Name | Name of the user on AIX computer that you want to use for connector operations. |
| gecos | The General Electric Comprehensive Operating System (GECOS) information for User. The user's name, phone numbers, and other generic personal information are stored here. |
| id | User ID |
| pgrp | Primary group of user. |
| groups | Secondary groups of user. |
| home | Home directory of user. |
| shell | Default shell of user. |
| login | Indicates whether the user can log in to the system with the login command. Possible values are: <ul style="list-style-type: none">■ true: The user can log in to the system. Default.■ false: The user cannot log in to the system. |
| su | Indicates whether another user can switch to the specified user account with the su command. Possible values are: <ul style="list-style-type: none">■ true: Another user can switch to the specified account. Default■ false: Another user cannot switch to the specified account. |
| rlogin | Permits access to the account from a remote location with the telnet or rlogin commands. Possible values are: <ul style="list-style-type: none">■ true: The user account can be accessed remotely. Default■ false: The user account cannot be accessed remotely. |

| Attributes | Description |
|------------|---|
| daemon | <p>Indicates whether the user specified by the <i>Name</i> parameter can execute programs using the cron daemon or the src (system resource controller) daemon. Possible values are:</p> <ul style="list-style-type: none"> ■ true: The user can initiate cron and src sessions. Default ■ false: The user cannot initiate cron and src sessions. |
| admin | <p>Defines the administrative status of the user. Possible values are:</p> <ul style="list-style-type: none"> ■ true: The user is an administrator. Only the root user can change the attributes of users defined as administrators. ■ false: The user is not an administrator. Default |
| dce_export | <p>Allows the DCE registry to overwrite the local user information with the DCE user information during a DCE export operation. Possible values are:</p> <ul style="list-style-type: none"> ■ true: Local user information will be overwritten ■ false: Local user information will not be overwritten |
| sugroups | <p>Lists the groups that can use the su command to switch to the specified user account. The <i>Value</i> parameter is a comma-separated list of group names, or a value of ALL to indicate all groups. An ! (exclamation point) in front of a group name excludes that group. If this attribute is not specified, all groups can switch to this user account with the su command.</p> |
| admgroups | <p>Lists the groups the user administrates. The <i>Value</i> parameter is a comma-separated list of group names. For additional information on group names, see the adms attribute of the /etc/security/group file.</p> |
| tpath | <p>Indicates the user's trusted path status. The possible values are:</p> <ul style="list-style-type: none"> ■ always: The user can only execute trusted processes. This implies that the user's initial program is in the trusted shell or some other trusted process. ■ notsh: The user cannot invoke the trusted shell on a trusted path. If the user enters the secure attention key (SAK) after logging in, the login session ends. ■ nosak: The secure attention key (SAK) is disabled for all processes run by the user. Use this value if the user transfers binary data that may contain the SAK sequence. Default ■ on: The user has normal trusted path characteristics and can invoke a trusted path (enter a trusted shell) with the secure attention key (SAK). |
| ttys | <p>Lists the terminals that can access the account specified by the <i>Name</i> parameter. The <i>Value</i> parameter is a comma-separated list of full path names, or a value of ALL to indicate all terminals. The values of RSH and REXEC also can be used as terminal names. An ! (exclamation point) in front of a terminal name excludes that terminal. If this attribute is not specified, all terminals can access the user account. If the <i>Value</i> parameter is not ALL, then /dev/pts must be specified for network logins to work.</p> |

Schema attributes

| Attributes | Description |
|--------------|---|
| expires | Identifies the expiration date of the account. The <i>Value</i> parameter is a 10-character string in the <i>MMDDhhmmyy</i> form, where <i>MM</i> = month, <i>DD</i> = day, <i>hh</i> = hour, <i>mm</i> = minute, and <i>yy</i> = last 2 digits of the years 1939 through 2038. All characters are numeric. If the <i>Value</i> parameter is 0, the account does not expire. The default is 0. See the date command for more information. |
| auth1 | <p>Lists additional mandatory methods for authenticating the user. The auth1 attribute has been deprecated and may not be supported in a future release. The SYSTEM attribute should be used instead. The authentication process will fail if any of the methods specified by the auth1 attribute fail.</p> <p>The <i>Value</i> parameter is a comma-separated list of <i>Method;Name</i> pairs. The <i>Method</i> parameter is the name of the authentication method. The <i>Name</i> parameter is the user to authenticate. If you do not specify a <i>Name</i> parameter, the name of the user being authenticated is used. Valid authentication methods for the auth1 and auth2 attributes are defined in the <i>/etc/security/login.cfg</i> file.</p> |
| auth2 | <p>Lists additional optional methods for authenticating the user. The auth2 attribute has been deprecated and may not be supported in a future release. The SYSTEM attribute should be used instead. The authentication process will not fail if any of the methods specified by the auth2 attribute fail.</p> <ul style="list-style-type: none"> ■ The <i>Value</i> parameter is a comma-separated list of <i>Method;Name</i> pairs. ■ The <i>Method</i> parameter is the name of the authentication method. ■ The <i>Name</i> parameter is the user to authenticate. If you do not specify a <i>Name</i> parameter, the name of the user being authenticated is used. |
| umask | Determines file permissions. This value, along with the permissions of the creating process, determines a file's permissions when the file is created. The default is 022. |
| registry | Defines the authentication registry where the user is administered. It is used to resolve a remotely administered user to the local administered domain. This situation may occur when network services unexpectedly fail or network databases are replicated locally. Example values are files or NIS or DCE. |
| loginretries | <p>Defines the number of unsuccessful login attempts allowed after the last successful login before the system locks the account. The value is a decimal integer string. A zero or negative value indicates that no limit exists. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's unsuccessful_login_count attribute in the <i>/etc/security/lastlog</i> file to be less than the value of loginretries. To do this, enter the following:</p> <pre>chsec -f /etc/security/lastlog -s username -a \ unsuccessful_login_count=0</pre> |
| pwdwarntime | Defines the number of days before the system issues a warning that a password change is required. The value is a decimal integer string. A zero or negative value indicates that no message is issued. The value must be less than the difference of the maxage and minage attributes. Values greater than this difference are ignored, and a message is issued when the minage value is reached. |

| Attributes | Description |
|----------------|--|
| account_locked | <p>Indicates if the user account is locked. Possible values include:</p> <ul style="list-style-type: none"> ■ true: The user's account is locked. The values yes, true, and always are equivalent. The user is denied access to the system. ■ false: The user's account is not locked. The values no, false, and never are equivalent. The user is allowed access to the system. Default |
| minage | <p>Defines the minimum age (in weeks) a password must be before it can be changed. The value is a decimal integer string. The default is a value of 0, indicating no minimum age.</p> |
| SYSTEM | <p>Defines the system authentication mechanism for the user. The value may be an expression describing which authentication methods are to be used or it may be the keyword NONE.</p> <p>The SYSTEM mechanism is always used to authenticate the user, regardless of the value of the auth1 and auth2 attributes. If the SYSTEM attribute is set to NONE, authentication is only performed using the auth1 and auth2 attributes. If the auth1 and auth2 attributes are blank or ignored, as with the TCP socket daemons (ftpd, rexecd and rshd), no authentication will be performed.</p> <p>The method names compat, files and NIS are provided by the security library. Additional methods may be defined in the /usr/lib/security/methods.cfg file.</p> <p>Specify the value for SYSTEM using the following grammar:</p> <pre> "SYSTEM" ::= EXPRESSION EXPRESSION ::= PRIMITIVE "(" EXPRESSION ")" EXPRESSION OPERATOR EXPRESSION PRIMITIVE ::= METHOD METHOD "[" RESULT "]" RESULT ::= "SUCCESS" "FAILURE" "NOTFOUND" "UNAVAIL" "*" OPERATOR ::= "AND" "OR" METHOD ::= "compat" "files" "NONE" [a-z,A-Z,0-9]* </pre> <p>An example of the syntax is: SYSTEM = "DCE OR DCE[UNAVAIL] AND compat"</p> |
| maxage | <p>Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is a decimal integer string. The default is a value of 0, indicating no maximum age.</p> |
| maxexpired | <p>Defines the maximum time (in weeks) beyond the maxage value that a user can change an expired password. After this defined time, only an administrative user can change the password. The value is a decimal integer string. The default is -1, indicating no restriction is set. If the maxexpired attribute is 0, the password expires when the maxage value is met. If the maxage attribute is 0, the maxexpired attribute is ignored.</p> |
| minalpha | <p>Defines the minimum number of alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.</p> |

Schema attributes

| Attributes | Description |
|------------|---|
| minother | Defines the minimum number of non-alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. |
| logintimes | <p>Specifies the times, days, or both, the user is allowed to access the system. The value is a comma-separated list of entries of the following form:</p> <pre>[!]:time-time -or- [!]day[-day][:time-time] -or- [!]date[-date][:time-time]</pre> <p>The <i>day</i> variable must be one digit between 0 and 6 that represents one of the days of the week. A 0 (zero) indicates Sunday and a 6 indicates Saturday.</p> <p>The <i>time</i> variable is 24-hour military time (1700 is 5:00 p.m.). Leading zeroes are required. For example, you must enter 0800, not 800. The <i>time</i> variable must be four characters in length, and there must be a leading colon (:). An entry consisting of only a time specification applies to every day. The start hour of a time value must be less than the end hour.</p> <p>The <i>date</i> variable is a four digit string in the form <i>mmdd</i>. <i>mm</i> represents the calendar month and <i>dd</i> represents the day number. For example 0001 represents January 1. <i>dd</i> may be 00 to indicate the entire month, if the entry is not a range, or indicating the first or last day of the month depending on whether it appears as part of the start or end of a range. For example, 0000 indicates the entire month of January. 0600 indicates the entire month of June. 0311-0500 indicates April 11 through the last day of June.</p> <p>Entries in this list specify times that a user is allowed or denied access to the system. Entries not preceded by an ! (exclamation point) allow access and are called ALLOW entries. Entries prefixed with an ! (exclamation point) deny access to the system and are called DENY entries. The ! operator applies to only one entry, not the whole restriction list. It must appear at the beginning of each entry.</p> |
| mindiff | Defines the minimum number of characters required in a new password that were not in the old password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. |
| maxrepeats | Defines the maximum number of times a character can be repeated in a new password. Since a value of 0 is meaningless, the default value of 8 indicates that there is no maximum number. The value is a decimal integer string. |
| minlen | Defines the minimum length of a password. The value is a decimal integer string. The default is a value of 0, indicating no minimum length. The maximum value allowed is 8. This attribute is determined by the minalpha attribute value added to the minother attribute value. If the sum of these values is greater than the minlen attribute value, the minimum length is set to the result. |
| histexpire | Designates the period of time (in weeks) that a user cannot reuse a password. The value is a decimal integer string. The default is 0, indicating that no time limit is set. |
| histsize | Designates the number of previous passwords a user cannot reuse. The value is a decimal integer string. The default is 0. |

| Attributes | Description |
|--------------------------|---|
| pwdchecks | Defines the password restriction methods enforced on new passwords. The value is a list of comma-separated method names and is evaluated from left to right. A method name is either an absolute path name or a path name relative to /usr/lib of an executable load module. |
| dictionlist | <p>Defines the password dictionaries used by the composition restrictions when checking new passwords.</p> <p>The password dictionaries are a list of comma-separated, absolute path names that are evaluated from left to right. All dictionary files and directories must be write-protected from all users except root. The dictionary files are formatted one word per line. The word begins in the first column and terminates with a new-line character. Only 7-bit ASCII words are supported for passwords. If text processing is installed on your system, the recommended dictionary file is the /usr/share/dict/words file.</p> |
| default_roles | Specifies the default roles for the user. The Value parameter, a comma-separated list of valid role names, can only contain roles assigned to the user in the roles attribute. You can use the ALL keyword to signify that the default roles for the user are all their assigned roles. |
| fsize | Identifies the soft limit for the largest file a user process can create or extend. |
| cpu | Sets the soft limit for the largest amount of system unit time (in seconds) that a user process can use. |
| data | Identifies the soft limit for the largest process data segment for a user process. |
| stack | Specifies the soft limit for the largest process stack segment for a user process. |
| core | Specifies the soft limit for the largest core file a user process can create. |
| rss | Sets the soft limit for the largest amount of physical memory a user process can allocate. This limit is not enforced by the system. |
| nofiles | Sets the soft limit for the number of file descriptors a user process may have open at one time. |
| stack_hard | Specifies the largest process stack segment for a user process. |
| roles | Contains the list of roles for each user. |
| time_last_login | Specifies the number of seconds since the epoch (00:00:00 GMT, January 1, 1970) since the last successful login. The value is a decimal integer. |
| tty_last_login | Specifies the terminal on which the user last logged in. The value is a character string. |
| host_last_login | Specifies the host from which the user last logged in. The value is a character string. |
| unsuccessful_login_count | <p>Specifies the number of unsuccessful login attempts since the last successful login. The value is a decimal integer. This attribute works in conjunction with the user's loginretries attribute, specified in the /etc/security/user file, to lock the user's account after a specified number of consecutive unsuccessful login attempts. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's unsuccessful_login_count attribute to be less than the value of loginretries. To do this, enter the following:</p> <pre>chsec -f /etc/security/lastlog -s username -a \ unsuccessful_login_count=0</pre> |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|------------|--|
| users | Identifies a list of one or more users which are associated with group. |
| admin | Specifies whether administrative group or not. |
| registry | Specifies where the user or group identification information is administrated. |
| Group Name | Name of group |
| id | Group ID |

Provisioning policy attributes

This section lists the different policy attributes of AIX Connector.

Account attributes

The following table lists the provisioning policy attributes for Create and Update Account:

| Attributes | Description |
|------------|--|
| User Name | <i>(Only for Create Account)</i> User ID on AIX computer that you want to use for connector operations. |
| id | User ID |
| pgrp | Primary group of user |
| home | Home directory of user |
| shell | Default shell of user |
| login | Indicates whether the user can log in to the system with the login command. Possible values are: <ul style="list-style-type: none">■ true: The user can log in to the system. Default.■ false: The user cannot log in to the system. |
| su | Indicates whether another user can switch to the specified user account with the su command. Possible values are: <ul style="list-style-type: none">■ true: Another user can switch to the specified account. Default■ false: Another user cannot switch to the specified account. |

| Attributes | Description |
|---------------|---|
| rlogin | Permits access to the account from a remote location with the telnet or rlogin commands. Possible values are: <ul style="list-style-type: none"> ■ true: The user account can be accessed remotely. Default ■ false: The user account cannot be accessed remotely. |
| admin | Defines the administrative status of the user. Possible values are: <ul style="list-style-type: none"> ■ true: The user is an administrator. Only the root user can change the attributes of users defined as administrators. ■ false: The user is not an administrator. Default |
| sugroups | Lists the groups that can use the su command to switch to the specified user account. The <i>Value</i> parameter is a comma-separated list of group names, or a value of ALL to indicate all groups. An ! (exclamation point) in front of a group name excludes that group. If this attribute is not specified, all groups can switch to this user account with the su command. |
| admgroups | Lists the groups the user administrates. The <i>Value</i> parameter is a comma-separated list of group names. For additional information on group names, see the adms attribute of the /etc/security/group file. |
| umask | Determines file permissions. This value, along with the permissions of the creating process, determines a file's permissions when the file is created. The default is 022. |
| default_roles | Specifies the default roles for the user. The <i>Value</i> parameter, a comma-separated list of valid role names, can only contain roles assigned to the user in the roles attribute. You can use the ALL keyword to signify that the default roles for the user are all their assigned roles. |

Group attributes

The following table lists the provisioning policy attributes for Create and Update Group:

| Attributes | Description |
|------------|---|
| Group Name | <i>(Only for create group)</i> Name of group |
| users | Identifies a list of one or more users which are associated with group. |
| Id | Group ID |

Additional information

This section describes the additional information related to the AIX Connector.

Unstructured Target Collector

AIX uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group **identityAttribute** for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint IdentityIQ User's Guide*.

For AIX target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

AIX Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated to IdentityIQ Users and Groups. For UNIX platforms direct access means ownership of file or directory.

Table 31—Unstructured Target Configuration parameters

| Attributes | Description | Possible values |
|---------------------------|--|--|
| Unix File System Path(s)* | Absolute path(s) which are to be scanned for resources. | Multiple paths can be mentioned with comma separated values. For example, /etc/tmp |
| Application Name* | Name of the application with which Unstructured Target will be correlated. | |

Note: Attributes marked with * sign are the mandatory attributes.

Note: If Unstructured Configuration is configured before upgrading IdentityIQ to version 6.1 from IdentityIQ version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

Correlation Rule: The rule used to determine how to correlate account and group information from the application with identity cubes in IdentityIQ.

Troubleshooting

1 - Aggregation fails on AIX 5.3 when number of users exceeds 1000

Aggregation fails on AIX version 5.3.

Resolution: Set the **LDR_CNTRL** environment variable as follows in run scripts of default shell for AIX connector administrator user:

```
export LDR_CNTRL =MAXDATA=0x60000000
```

Increase the number of stack segments from 1 to 8 depending on the number of Users, Groups and Connections present on the AIX computer.

MAXDATA=0xN0000000@DSA

where *N* is the number of stack segments.

2 - Test connection fails for managed systems

Test Connection fails for managed systems with the following error when SSH login prompt appears with some delay:

Test Connection failed. Login failed. 'sh' is not set on your machine. Please set 'sh'

The above error occurs when connector tries to login to target managed system with SSH and execute the **sh** command. The **sh** command fails because of delay on target managed system for SSHLogin prompt to appear.

Resolution: To resolve this issue, tune the following time out parameters according to your need in the Application Debug page:

- **SSHLoginTimeout:** Default value: 1000 ms
This time out parameter is responsible to tune time taken to connect to target host through ssh.
This is also effective in tuning time taken between actual login process start and actual appearance of first prompt.
- **SSHTimeOut:** Default value: 120000 ms
This time out parameter is responsible to tune maximum time for which a ssh command execution should be allowed. After this time out even if the command execution is in progress on target host, the connection will be dropped out and the operation will be timed out.

3 - Aggregation/test connection fails with timeout error

Aggregation/test connection fails with the following timeout error:

Exception during aggregation. Reason: sailpoint.connector.ConnectorException: Account aggregation failed. Timeout occurred.

Resolution: Change the value of the **SSHLoginTimeout (in millisecond)** application attribute as per your requirement in the debug page of the application:

```
<entry key="SSHLoginTimeout" value="1000"/>
```

4 - After target aggregation resources are not getting correlated with Account Groups.

After target aggregation the resources are not getting correlated with Account Groups.

Resolution: Ensure that your correlation rule populates "Correlator.RULE_RETURN_GROUP_ATTRIBUTE" as follows:

```
....
if ( isGroup ) {
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE,"nativeIdentity");
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);
}
....
```

5 - Test connection fails for key based authentication with an error.

Test connection fails for key based authentication with the following error.

Login failed. Error while connecting to host:<hostname>. Cannot read key file.

Troubleshooting

Resolution: Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using openssl. This method can be used for any version of SSH.
 - a. Create private key using the following command:
openssl <genssa/genrsa> -des3 -out <private_key> 1024
 - b. Change the permission on the <private_key> file as follows:
chmod 0600 <private_key>
 - c. Create public key from private_key
ssh-keygen -y -f <private_key> > <public_key>
 - d. Use the <private_key> and <public_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
 - a. Create private and public key using the following command
ssh-keygen -t <dsa/rsa> -b 1024

By default files with name **id_dsa/id_rsa** and **id_dsa.pub/id_rsa.pub** will be created.
 - b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:
openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3
 - c. Change the permission on the <new_private_key> file as follows:
chmod 0600 <new_private_key>
 - d. Create public key file using the new private key as follows:
ssh-keygen -y -f <new_private_key> > <new_public_key>
 - e. Use the <new_private_key> and <new_public_key> files for authentication.

6 - Test connection fails with an error when sudo user is configured for public key authentication

Test connection fails with the following error when sudo user is configured for public key authentication:

Test SSH communication failed over host: xxxxxxxx. Error while executing command: sudo -p %SAILPOINTSUDO echo TestConnection over host: xxxxxxxx. Invalid sudo user password.

Resolution: Verify the sudo user's password specified in application configuration, password should be correct for certificate based authentication.

Chapter 23: SailPoint IdentityIQ Linux Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 197 |
| Supported features | 197 |
| Supported Managed Systems | 198 |
| Pre-requisites | 198 |
| Administrator permissions | 198 |
| Configuration parameters | 198 |
| Additional configuration parameters for SSH configuration | 199 |
| Public key authentication configuration | 199 |
| Schema attributes | 200 |
| Account attributes | 200 |
| Group attributes | 201 |
| Provisioning policy attributes | 201 |
| Account attributes | 201 |
| Group attributes | 202 |
| Additional information | 202 |
| Unstructured Target Collector | 202 |
| Troubleshooting | 203 |

Overview

Linux Connector was developed to enable user managing their Linux Account, Groups and resources from IdentityIQ. The Linux system data will be aggregated to IdentityIQ and user would be able to edit entities and their attributes.

Supported features

The Linux Connector provides support for the following features:

- Account Aggregation
 - Account Group Aggregation
 - Create/Update/Delete Account
 - Get/Sync Account
 - Enable/Disable/Unlock Account
 - Change Password
 - Create/Update/Delete Account Group
 - Add/Delete entitlement
 - Reset password
 - Target Aggregation
- For more information, see [“Additional information” on page 202](#).
- Revoke Target Permissions
 - Password Interceptor

Configuration parameters

Password Interceptor for Linux provides the mechanism by which a password change initiated from Linux system is captured by the Client and sent to IdentityIQ. For more information, see [“Password Interceptor for UNIX”](#) section in [Appendix , “A: Password Interceptor”](#).

Supported Managed Systems

The Linux connector supports the following versions of the operating system:

- Red Hat Enterprise Linux versions 5.8, 6.0, 6.1, 6.2, and 6.3
- SUSE Linux version 10, 11 SP1, and 11 SP2

Note: For any issues related to SUSE Linux, see [“Troubleshooting”](#) on page 203 section.

Pre-requisites

SSH should be installed on Linux computer.

Administrator permissions

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

Rights to execute the following commands with root privilege:

```
/bin/chmod, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel,  
/usr/sbin/groupadd, /usr/sbin/groupmod, /usr/sbin/groupdel,  
/usr/bin/passwd, /usr/bin/faillog, /usr/bin/groups, /bin/rm, /bin/echo,  
/usr/bin/chage, /usr/bin/find, /bin/cat /etc/shadow, /bin/cat  
/etc/passwd, /bin/cat /etc/group, /bin/cat /etc/pam.d/system-auth
```

An entry in /etc/sudoers file should look similar to the following:

```
username ALL = (root) PASSWD : /bin/chmod, /usr/sbin/useradd,  
/usr/sbin/usermod, /usr/sbin/userdel, /usr/sbin/groupadd,  
/usr/sbin/groupmod, /usr/sbin/groupdel, /usr/bin/passwd,  
/usr/bin/faillog, /usr/bin/groups, /bin/rm, /bin/echo, /usr/bin/chage,  
/usr/bin/find, /bin/cat /etc/shadow, /bin/cat /etc/passwd, /bin/cat  
/etc/group, /bin/cat /etc/pam.d/system-auth
```

Note: All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on Linux computers as they might differ from the values mentioned here.

Note: If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

Configuration parameters

The following table lists the configuration parameters of Linux Connector:

| Parameters | Description |
|----------------------------|--|
| Unix Server Host | Host Name/IP address of the computer. |
| SSH Port | SSH port configured. Default value: 22 |
| User Name | User ID on the computer that you want to use for connector operations. |
| User Password | Password of the target system user account that you want to use for connector operations. |
| Not a 'root' user | If User ID specified is not root, check this paramter. |
| Private Key File Path | Path to Private Key File. Private/Public key authentication will have precedence over password authentication. |
| Passphrase For Private Key | Passphrase provided for creating Private Key. |

Additional configuration parameters for SSH configuration

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

Note: These additional configuration parameters must be added in the Application/Target Source debug page.

- Following is the default command for setting shell prompt on UNIX computer:

```
<entry key="SetPrompt" value="PS1='SAILPOINT>'"/>
```

In the above command, “SetPrompt” is the application/target source attribute and PS1=‘SAILPOINT’ is the value of the application/target source attribute.

If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

For example: For tcsh shell, the entry value would be:

```
<entry key="SetPrompt" value="set prompt='SAILPOINT>'"/>
```

- For executing the commands, verify that the default shell is present on your system. If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

```
<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>
```

Public key authentication configuration

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

Schema attributes

1. Generate Private and Public key's. For more information of the standard steps, see [“8 - Test connection fails for key based authentication with an error.” on page 205.](#)
2. Append contents of public key file to `~/.ssh/authorized_keys` as shown below.
`cat <public key file> >> ~/.ssh/authorized_keys`
3. Copy private key file to a location which is accessible by IdentityIQ server.
4. Provide path of private key file in application configuration.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|------------|---|
| username | It is used when user logs in. |
| uid | Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups. |
| home | The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes / |
| pwdlastchg | Days since Jan 1, 1970 that password was last changed. |
| pwdmin | The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password. |
| pwdmax | The maximum number of days the password is valid (after that user is forced to change his/her password). |
| pwdwarn | The number of days before password is to expire that user is warned that his/her password must be changed. |
| comment | Description |
| expiration | Days since Jan 1, 1970 that account is disabled that is, an absolute date specifying when the login may no longer be used. |
| inactive | The number of days after password expires that account is disabled. |
| lastlogin | Last login date and time of the Account. |
| primgrp | Name of primary group of the user. |
| shell | User's shell. |
| groups | Secondary groups of user. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|------------|--|
| groupid | GID. Each user must be assigned a group ID. You can see this number in your <code>/etc/group</code> file. |
| name | It is the name of group. If you run <code>ls -l</code> command, you will see this name printed in the group field. |

Provisioning policy attributes

This section lists the different policy attributes of Linux Connector.

Account attributes

The following table lists the provisioning policy attributes for Create Account:

| Attributes | Description |
|------------------------------|---|
| User Name | It is used when user logs in. It should be between 1 and 32 characters in length. |
| User ID | Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups. |
| Home Directory | The absolute path to the directory the user will be in when they log in. If this directory does not exist then user's directory becomes / |
| Min password change days | The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password. |
| Max password validity | The maximum number of days the password is valid (after that user is forced to change his/her password). |
| Password change warning time | The number of days before password is to expire that user is warned that his/her password must be changed. |
| Comment | Description |
| Account expire duration | Days since Jan 1, 1970 that account is disabled that is, an absolute date specifying when the login may no longer be used. |
| Account inactivity time | The number of days after password expires that account is disabled. |
| Do not add to last login | Whether to add to last login log file. |
| Shell | User's shell. |
| Allow duplicate UID | Allow creation of account with a duplicate (non-unique) UID. |

Additional information

| Attributes | Description |
|-----------------------|--|
| Create Home Directory | Whether to create home directory for new user. |
| Primary Group name | Specify primary group name. |

Group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|---------------------|--|
| Group ID | GID. Each user must be assigned a group ID. You can see this number in your <code>/etc/passwd</code> file. |
| Group Name | It is the name of group. If you run <code>ls -l</code> command, you will see this name printed in the group field. |
| Allow duplicate GID | Duplicate GID of Group. |

Additional information

This section describes the additional information related to the Linux Connector.

Unstructured Target Collector

Linux uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group **identityAttribute** for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint IdentityIQ User's Guide*.

For Linux target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

Linux Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated to IdentityIQ Users and Groups. For UNIX platforms direct access means ownership of file or directory.

| Attributes | Description | Possible values |
|---------------------------|---|---|
| Unix File System Path(s)* | Absolute path(s) which are to be scanned for resources. | Multiple paths can be mentioned with comma separated values. For example, <code>/etc,/tmp</code> |

| Attributes | Description | Possible values |
|-------------------|--|-----------------|
| Application Name* | Name of the application with which Unstructured Target will be correlated. | |

Note: Attributes marked with * sign are the mandatory attributes.

Note: If Unstructured Configuration is configured before upgrading IdentityIQ to version 6.1 from IdentityIQ version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

Correlation Rule: The rule used to determine how to correlate account and group information from the application with identity cubes in IdentityIQ.

Troubleshooting

1 - Test connection failed on SUSE computer with an error message.

Test connection failed on SUSE computer with the following error message:

Unexpected output captured from host: 172.16.22.124. Expected: TestConnection.

Captured: sword sudo: pam_authenticate: Module is unknown SAILPOINT> Password

Sh : Password : command not found. Command exit code: sword sudo:

pam_authenticate: Module is unknown SAILPOINT> Password sh: Password: command not found

Resolution: When the test connection fails on SUSE computer, the following setting must be changed in `/etc/ssh/sshd_config` file:

```
PasswordAuthentication yes
```

Enter the following command to restart the `sshd` after updating the `sshd_config` file:

```
/etc/init.d/sshd restart
```

2 - After Account Aggregation Lock/Unlock status is not displayed correctly on IdentityIQ.

After Account Aggregation the Lock/Unlock status is not displayed correctly on IdentityIQ.

Resolution: Ensure that faillog command, as required in the following registry key works correctly:

```
<entry key="aggregation.lockstatus" value="faillog | awk '{print $1} {print $2}'"/>
```

3 - Refresh account does not set correct Lock/Unlock status on IdentityIQ.

Resolution: Perform the following:

1. Specify the maximum allowed failed login attempts before the account is locked by the system. Edit the configuration file pointed by registry key:

```
<entry key="get.loginsyslimit" value="cat /etc/pam.d/system-auth"/>
```

Troubleshooting

Default value: `/etc/pam.d/system-auth`

Specify maximum allowed failed login using `"deny="`.

For example, add the following lines in `/etc/pam.d/system-auth`:

`auth required pam_tally2.so onerr=fail deny=5`

`account required pam_tally2.so`

2. Ensure that the following command to get failed login works on the system:
`<entry key="get.userfailedlogin" value="faillog"/>`

4 - Unlock account does not work.

Resolution: Verify if unlock command given in the registry correctly resets the failed login counter. Default settings: `<entry key="unlock.account" value="faillog -u"/>`

5 - Password command failed with an error message.

Password command fails if password prompts are not matching.

Resolution: Verify the password command on Linux computer for password prompts and if the required prompts are present in your application.

For example, passwd Person2

Changing password for Person2.

New Password: New Password is the prompt, so if this prompt is not present in your application, add/update it as follows:

For example,

```
<entry key="PasswdPrompts">
  <value>
    <Map>
      <entry key="0">
        <value>
          <Map>
            <entry key="(current) UNIX password:" value="CurrentPassword"/>
          </Map>
        </value>
      </entry>
      <entry key="1">
        <value>
          <Map>
            <entry key="Old Password:" value="CurrentPassword"/>
          </Map>
        </value>
      </entry>
      <entry key="2">
        <value>
          <Map>
            <entry key="New Password:" value="NewPassword"/>
          </Map>
        </value>
      </entry>
    </Map>
  </value>
</entry>
```

...
...
...
...
...
...
...
...

6 - Aggregation/test connection fails with timeout error

Aggregation/test connection fails with the following timeout error:

Exception during aggregation. Reason: sailpoint.connector.ConnectorException: Account aggregation failed. Timeout occurred.

Resolution: Change the value of the **SSHLoginTimeout (in millisecond)** application attribute as per your requirement in the debug page of the application:

```
<entry key="SSHLoginTimeout" value="1000"/>
```

7 - After target aggregation resources are not getting correlated with Account Groups.

After target aggregation the resources are not getting correlated with Account Groups.

Resolution: Ensure that your correlation rule populates "Correlator.RULE_RETURN_GROUP_ATTRIBUTE" as follows:

```
.....
if ( isGroup ) {
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE,"nativeIdentity");
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);
}
.....
```

8 - Test connection fails for key based authentication with an error.

Test connection fails for key based authentication with the following error.

Login failed. Error while connecting to host:<hostname>. Cannot read key file.

Resolution: Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using openssl. This method can be used for any version of SSH.
 - a. Create private key using the following command:


```
openssl <genssa/genrsa> -des3 -out <private_key> 1024
```
 - b. Change the permission on the <private_key> file as follows:


```
chmod 0600 <private_key>
```
 - c. Create public key from private_key


```
ssh-keygen -y -f <private_key> > <public_key>
```
 - d. Use the <private_key> and <public_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
 - a. Create private and public key using the following command

Troubleshooting

ssh-keygen -t <dsa/rsa> -b 1024

By default files with name **id_dsa/id_rsa** and **id_dsa.pub/id_rsa.pub** will be created.

- b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:

openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3

- c. Change the permission on the <new_private_key> file as follows:

chmod 0600 <new_private_key>

- d. Create public key file using the new private key as follows:

ssh-keygen -y -f <new_private_key> > <new_public_key>

- e. Use the <new_private_key> and <new_public_key> files for authentication.

9 - Test connection fails with an error when sudo user is configured for public key authentication

Test connection fails with the following error when sudo user is configured for public key authentication:

Test SSH communication failed over host: xxxxxxxx. Error while executing command: sudo -p %SAILPOINTSUDO echo TestConnection over host: xxxxxxxx. Invalid sudo user password.

Resolution: Verify the sudo user's password specified in application configuration, password should be correct for certificate based authentication.

Chapter 24: SailPoint IdentityIQ Solaris Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 207 |
| Supported features | 207 |
| Supported Managed Systems | 208 |
| Pre-requisites | 208 |
| Administrator permissions | 208 |
| Configuration parameters | 209 |
| Additional configuration parameters for SSH configuration | 209 |
| Public key authentication configuration | 209 |
| Schema attributes | 210 |
| Account attributes | 210 |
| Group attributes | 212 |
| Provisioning policy attributes | 212 |
| Account attributes | 212 |
| Group attributes | 213 |
| Additional information | 213 |
| Unstructured Target Collector | 214 |
| Troubleshooting | 214 |

Overview

In Solaris Connector users on Solaris computer are used for account provisioning. For group provisioning, groups are used. You can configure the Connector to use any of the attributes of user/group which are supported by Solaris commands.

Supported features

The Solaris Connector provides support for the following features:

- Account Aggregation
- Account Group Aggregation
- Create/Delete/Update Account
- Enable/Disable/Unlock Account
- Get/Sync Account
- Change Password
- Create/Update/Delete Account Group
- Add/Delete Entitlement
- Reset password
- Target Aggregation
- For more information, see [“Unstructured Target Collector”](#) on page 214.
- Revoke Target Permissions
- Password Interceptor

Password Interceptor for Solaris provides the mechanism by which a password change initiated from Solaris system is captured by the Client and sent to IdentityIQ. For more information, see “Password Interceptor for UNIX” section in [Appendix , “A: Password Interceptor”](#).

Note: Role provisioning is not supported by Solaris Connector.

Supported Managed Systems

The Solaris connector supports the following versions of the operating system:

- Solaris 9 SPARC x86
- Solaris 10 SPARC x86
- Solaris 11 SPARC x86

Note: For any issues related to Solaris, see “[Troubleshooting](#)” on [page 214](#) section.

Pre-requisites

SSH should be installed on Solaris computer.

Administrator permissions

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

Rights to execute the following commands with root privilege:

```
/bin/chmod, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel,
/usr/sbin/groupadd, /usr/sbin/groupmod, /usr/sbin/groupdel,
/usr/bin/passwd, /usr/bin/groups, /usr/bin/date, /bin/rm, /bin/echo,
/usr/bin/find, /bin/cat /etc/shadow, /bin/cat /etc/passwd, /bin/cat
/etc/group, /bin/cat /etc/user_attr, /bin/grep -i 'RETRIES='
/etc/default/login, /bin/grep -i 'Lock_After_Retries='
/etc/security/policy.conf
```

An entry in /etc/sudoers file should look similar to the following:

```
username ALL = (root) PASSWD : /bin/chmod, /usr/sbin/useradd,
/usr/sbin/usermod, /usr/sbin/userdel, /usr/sbin/groupadd,
/usr/sbin/groupmod, /usr/sbin/groupdel, /usr/bin/passwd,
/usr/bin/groups, /usr/bin/date, /bin/rm, /bin/echo, /usr/bin/find,
/bin/cat /etc/shadow, /bin/cat /etc/passwd, /bin/cat /etc/group, /bin/cat
/etc/user_attr, /bin/grep -i 'RETRIES=' /etc/default/login, /bin/grep -i
'Lock_After_Retries=' /etc/security/policy.conf
```

Note: All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on Solaris computers as they might differ from the values mentioned here.

Note: If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

Configuration parameters

The following table lists the configuration parameters of Solaris Connector:

| Parameters | Description |
|----------------------------|--|
| UNIX Server Host | Host Name/IP address of Solaris computer. |
| SSH Port | SSH port configured. Default value: 22 |
| Not a 'root' user | If User ID specified is not root, check this paramter. |
| User Name | User ID on Solaris computer that you want to use for connector operations. |
| User Password | Password of the target system user account that you want to use for connector operations. Default value: sadmin |
| Private Key File Path | Path to Private Key File. Private/Public key authentication will have precedence over password authentication. |
| Passphrase For Private Key | Passphrase provided for creating Private Key. |

Additional configuration parameters for SSH configuration

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

Note: These additional configuration parameters must be added in the Application/Target Source debug page.

- Following is the default command for setting shell prompt on UNIX computer:

```
<entry key="SetPrompt" value="PS1='SAILPOINT>'"/>
```

In the above command, "SetPrompt" is the application/target source attribute and PS1='SAILPOINT' is the value of the application/target source attribute.

If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

For example: For tcsh shell, the entry value would be:

```
<entry key="SetPrompt" value="set prompt='SAILPOINT>'"/>
```

- For executing the commands, verify that the default shell is present on your system. If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

```
<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>
```

Public key authentication configuration

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Schema attributes

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

1. Generate Private and Public key's. For more information of the standard steps, see [“6 - Test connection fails for key based authentication with an error.” on page 216.](#)
2. Append contents of public key file to `~/.ssh/authorized_keys` as shown below.

```
cat <public key file> >> ~/.ssh/authorized_keys
```
3. Copy private key file to a location which is accessible by IdentityIQ server.
4. Provide path of private key file in application configuration.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|---------------|---|
| username | Name of user. |
| uid | Numeric ID for user. |
| primgrp | An existing group integer ID or character-string name. Without the -D option, it defines the new user primary group membership and defaults to the default group. You can reset this default value by invoking <code>useradd -D -g group</code> . GIDs 0-99 are reserved for allocation by the Solaris Operating System. |
| groups | Secondary groups of user. List of groups assigned to user. |
| roles | Contains the list of roles for each user. |
| home | Home directory of user. |
| shell | Default shell of user. |
| comment | Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's /etc/passwd entry. |
| authorization | One or more comma separated authorizations defined in <code>auth_attr(4)</code> . Only a user or role who has grant rights to the authorization can assign it to an account. |
| skel_dir | A directory that contains skeleton information (such as <code>profile</code>) that can be copied into a new user's home directory. This directory must already exist. The system provides the /etc/skel directory that can be used for this purpose |
| project | Name of the project with which the added user is associated. See the <code>projname</code> field as defined in <code>project(4)</code> . |

| Attributes | Description |
|--------------------|--|
| expire | Specify the expiration date for a login. After this date, no user will be able to access this login. The expire option argument is a date entered using one of the date formats included in the template file /etc/datemsk . See <code>getdate(3C)</code> . If the date format that you choose includes spaces, it must be quoted. For example, you can enter <code>10/6/90</code> or <code>October 6, 1990</code> . A null value (" ") defeats the status of the expired date. This option is useful for creating temporary logins. |
| inactive | The maximum number of days allowed between uses of a login ID before that ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status. |
| lock_after_retries | Specifies whether an account is locked after the count of failed logins for a user equals or exceeds the allowed number of retries as defined by <code>RETRIES</code> in /etc/default/login . Possible values are yes or no . The default is no . Account locking is applicable only to local accounts and accounts in the LDAP name service repository if configured with an enableShadowUpdate of true as specified in Idapclient(1M) . |
| limitpriv | The maximum set of privileges a user or any process started by the user, whether through <code>su(1M)</code> or any other means, can obtain. The system administrator must take ensure that when deleting the privileges from the limit set. Deleting any basic privilege has the ability of crippling all applications; deleting any other privilege can cause many or all applications requiring privileges to malfunction. |
| defaultpriv | The default set of privileges assigned to a user's inheritable set upon login. |
| profiles | Contains an ordered, comma-separated list of profile names selected from <code>prof_attr(4)</code> . Profiles are enforced by the profile shells, <code>pfsh</code> , <code>pfksh</code> , and <code>pfsh</code> . See <code>pfsh(1)</code> . A default profile is assigned in /etc/security/policy.conf (see <code>policy.conf(4)</code>). If no profiles are assigned, the profile shells do not allow the user to execute any commands. |
| failedretries | Indicates if the user account is locked. Possible values include: <ul style="list-style-type: none"> ■ true: The user account is locked. The values yes, true, and always are equivalent. The user is denied access to the system. ■ false: The user account is not locked. The values no, false, and never are equivalent. The user is allowed access to the system. Default value. |
| pwdminage | The minimum number of days required between password changes for user. <code>MINWEEKS</code> is found in /etc/default/passwd and is set to <code>NULL</code> . |
| pwdmaxage | The maximum number of days the password is valid for user. <code>MAXWEEKS</code> is found in /etc/default/passwd and is set to <code>NULL</code> . |
| pwdwarn | The number of days relative to max before the password expires and the name are warned. |
| pwdlastchg | The date password was last changed for name. All password aging dates are determined using Greenwich Mean Time (Universal Time) and therefore can differ by as much as a day in other time zones. |
| audit_flags | Specifies per-user Audit pre selection flags as colon-separated always-audit-flags and never-audit-flags . For example, <code>audit_flags=always-audit-flags:never-audit-flags</code> . |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|------------|-----------------------------|
| groupname | Name of the account group |
| groupid | Numeric ID of account group |

Provisioning policy attributes

This section lists the different policy attributes of Solaris Connector.

Account attributes

The following table lists the provisioning policy attributes for Create and Update Account:

| Attributes | Description |
|-----------------------|--|
| Create Account | |
| username | Name of user. |
| uid | Numeric ID for user. |
| | Allow duplication of User ID |
| primgrp | An existing group integer ID or character-string name. Without the -D option, it defines the new user primary group membership and defaults to the default group. You can reset this default value by invoking <code>useradd -D -g group</code> . GIDs 0-99 are reserved for allocation by the Solaris Operating System. |
| home | Home directory of user. |
| shell | Default shell of user. |
| comment | Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's /etc/passwd entry. |
| authorization | One or more comma separated authorizations defined in <code>auth_attr(4)</code> . Only a user or role who has grant rights to the authorization can assign it to an account. |
| profiles | Contains an ordered, comma-separated list of profile names selected from <code>prof_attr(4)</code> . Profiles are enforced by the profile shells, <code>pfsh</code> , <code>pfksh</code> , and <code>pfsh</code> . See <code>pfsh(1)</code> . A default profile is assigned in /etc/security/policy.conf (see <code>policy.conf(4)</code>). If no profiles are assigned, the profile shells do not allow the user to execute any commands. |
| project | Name of the project with which the added user is associated. See the <code>projname</code> field as defined in <code>project(4)</code> . |

| Attributes | Description |
|--------------------|--|
| expire | Specify the expiration date for a login. After this date, no user will be able to access this login. The expire option argument is a date entered using one of the date formats included in the template file <code>/etc/datemask</code> . See <code>getdate(3C)</code> . If the date format that you choose includes spaces, it must be quoted. For example, you can enter 10/6/90 or October 6, 1990. A null value (" ") defeats the status of the expired date. This option is useful for creating temporary logins. |
| inactive | The maximum number of days allowed between uses of a login ID before that ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status. |
| lock_after_retries | Specifies whether an account is locked after the count of failed logins for a user equals or exceeds the allowed number of retries as defined by <code>RETRIES</code> in <code>/etc/default/login</code> . Possible values are yes or no . The default is no . Account locking is applicable only to local accounts and accounts in the LDAP name service repository if configured with an enableShadowUpdate of true as specified in <code>ldapclient(1M)</code> . |
| pwdwarn | Warning period for user's password expiry. |
| pwdminage | Minimum period between user's password change. |
| forcepwdchange | If user has to be forced to change password on next logon. |
| pwdmaxage | Maximum period for which password is valid for user. |

Group attributes

The following table lists the provisioning policy attributes for Create and Update Group:

| Attributes | Description |
|--------------|-------------------------------|
| groupname | Name of the account group |
| groupid | Numeric ID of account group |
| dupgid | Allow duplication of groupid. |
| Update Group | |
| groupid | Numeric ID of account group. |
| dupgid | Allow duplication of groupid. |

Additional information

This section describes the additional information related to the Solaris Connector.

Unstructured Target Collector

Solaris uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group **identityAttribute** for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint IdentityIQ User's Guide*.

For Solaris target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

Solaris Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated to IdentityIQ Users and Groups. For UNIX platforms direct access means ownership of file or directory.

| Attributes | Description | Possible values |
|---------------------------|--|---|
| Unix File System Path(s)* | Absolute path(s) which are to be scanned for resources. | Multiple paths can be mentioned with comma separated values. For example, /etc,/tmp |
| Application Name* | Name of the application with which Unstructured Target will be correlated. | |

Note: Attributes marked with * sign are the mandatory attributes.

Note: If Unstructured Configuration is configured before upgrading IdentityIQ to version 6.1 from IdentityIQ version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

Correlation Rule: The rule used to determine how to correlate account and group information from the application with identity cubes in IdentityIQ.

Troubleshooting

1 - Test connection fails with an error.

The following error message appears when test connection fails:

```
java.io.IOException: Corrupt Mac on input
```

OR

```
Error: Login failed. Error while connecting to host: xxxxx. The message store has reached EOF
```

Resolution: Add Cipher **3des-cbc** or **blowfish-cbc** to the list of Cipher's in `/etc/ssh/sshd_config` file and restart `sshd`.

- **For X86:** include **3des-cbc** or **blowfish-cbc** in Ciphers list
For example, Ciphers aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc, arcfour, arcfour128, arcfour256, 3des-cbc, blowfish-cbc
- **For SPARC:** include **3des-cbc** in Ciphers list
For example, Ciphers aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc, arcfour, arcfour128, arcfour256, 3des-cbc

2 - Test connection fails with an error.

The following error message appears when test connection fails:

Login failed. Failed to authenticate the ssh credentials for user: root to host: xxxxxx

Resolution: Update `/etc/ssh/sshd_config` file for the following entry and restart `sshd`:

PasswordAuthentication yes

3 - Aggregation fails with an error.

The following error message appears when aggregation fails:

Exception during aggregation. Reason: sailpoint.connector.ConnectorException: Failed to execute command: cat /etc/group | grep -v '^+' | grep -v '^-' Error:

Resolution: Create home directory for sudo user and run aggregation again. Ensure that the sudo user is able to create files in its home directory.

4 - Test connection fails with an error.

The following error message appears when aggregation fails:

Fails with error Login failed. Failed to authenticate the ssh credentials for user: test to host: xxxxxx

Resolution: The **ksh93** shell is the default shell `/usr/sbin/sh -> ../bin/i86/ksh93`. The **J2SSH** library does not work properly with this shell.

In default installation of Solaris 11, bash and tcsh are installed, use one of them for provisioning. Use application attribute **DEFAULT_SSH_SHELL**.

For more information on **DEFAULT_SSH_SHELL** parameter, see [“Additional configuration parameters for SSH configuration” on page 209](#).

5 - Aggregation/test connection fails with timeout error.

Aggregation/test connection fails with the following timeout error:

Exception during aggregation. Reason: sailpoint.connector.ConnectorException: Account aggregation failed. Timeout occurred.

Resolution: Change the value of the **SSHLoginTimeout (in millisecond)** application attribute as per your requirement in the debug page of the application:

`<entry key="SSHLoginTimeout" value="1000"/>`

6 - After target aggregation resources are not getting correlated with Account Groups.

After target aggregation the resources are not getting correlated with Account Groups.

Resolution: Ensure that your correlation rule populates "Correlator.RULE_RETURN_GROUP_ATTRIBUTE" as follows:

```
.....
if (isGroup) {
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE,"nativeIdentity");
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);
}
.....
```

6 - Test connection fails for key based authentication with an error.

Test connection fails for key based authentication with the following error.

Login failed. Error while connecting to host:<hostname>. Cannot read key file.

Resolution: Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using openssl. This method can be used for any version of SSH.
 - a. Create private key using the following command:


```
openssl <gendsa/genrsa> -des3 -out <private_key> 1024
```
 - b. Change the permission on the <private_key> file as follows:


```
chmod 0600 <private_key>
```
 - c. Create public key from private_key


```
ssh-keygen -y -f <private_key> > <public_key>
```
 - d. Use the <private_key> and <public_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
 - a. Create private and public key using the following command


```
ssh-keygen -t <dsa/rsa> -b 1024
```

By default files with name **id_dsa/id_rsa** and **id_dsa.pub/id_rsa.pub** will be created.
 - b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:


```
openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3
```
 - c. Change the permission on the <new_private_key> file as follows:


```
chmod 0600 <new_private_key>
```
 - d. Create public key file using the new private key as follows:


```
ssh-keygen -y -f <new_private_key> > <new_public_key>
```
 - e. Use the <new_private_key> and <new_public_key> files for authentication.

7 - Test connection fails with an error when sudo user is configured for public key authentication

Test connection fails with the following error when sudo user is configured for public key authentication:

Test SSH communication failed over host: xxxxxxxx. Error while executing command: sudo -p %SAILPOINTSUDO echo TestConnection over host: xxxxxxxx. Invalid sudo user password.

Resolution: Verify the sudo user's password specified in application configuration, password should be correct for certificate based authentication.

Chapter 25: SailPoint IdentityIQ BMC Remedy IT Service Management Suite Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 219 |
| Supported features | 219 |
| Supported Managed Systems | 220 |
| Pre-requisites | 220 |
| Administrator permission | 220 |
| Configuration parameters | 220 |
| Schema attributes | 220 |
| Account attributes | 220 |
| Group attributes | 221 |
| Provisioning policy attributes | 222 |
| Create account attributes | 222 |
| Create group attributes | 223 |
| Update policies | 223 |
| Additional information | 224 |
| Enable/Disable Account | 224 |
| Add Entitlement operation for ITSM | 224 |
| Troubleshooting | 224 |

Overview

SailPoint IdentityIQ BMC Remedy IT Service Management Suite Connector was developed to manage the accounts and groups contained in a BMC Remedy IT Service Management Suite (ITSM).

Supported features

SailPoint IdentityIQ BMC Remedy ITSM Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Account Refresh
- Create/Delete/Update Account
- Enable/Disable Account
For more information, see [“Enable/Disable Account” on page 224](#).
- Add/Remove Entitlement
For more information, see [“Add Entitlement operation for ITSM” on page 224](#).
- Create/Update/Delete Account-Group
- Change Password
- Pass through Authentication

Note: Remedy ITSM connector supports only Support Groups or SGPs.

Supported Managed Systems

- BMC Remedy IT Service Management Suite version 8.1
- BMC Remedy IT Service Management Suite version 8.0
- BMC Remedy IT Service Management Suite version 7.6.04
- BMC Remedy IT Service Management Suite version 7.5.01

Pre-requisites

1. You must copy the **arapi<v>.jar** file from the location where the server is installed (**installFolder\BMC Software\ARSystem\midtier\WEB-INF\lib**) to the lib folder of IdentityIQ installation (**\webapps\identityiq\WEB-INF\lib**).
2. Add the location of **arapi<v>.jar** file to the CLASSPATH system variable (**\webapps\identityiq\WEB-INF\lib\arapi<v>.jar**) of the computer where IdentityIQ installed.
3. Provide the appropriate read and write permissions to the Administrator to perform the user and group provisioning operations from IdentityIQ.

Administrator permission

The Application User should be a member of the **Administrator** group.

Configuration parameters

The following table lists the configuration parameters of Remedy ITSM Connector:

| Parameters | Description |
|----------------------------------|--|
| Remedy Server name or IP Address | IP address of the computer on which the Remedy ITSM server is installed. |
| Administrator Name | Name of the Remedy ITSM administrator. |
| Administrator Password | Password of the administrator. |
| Server Port | Remedy ITSM Server port number. |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|---------------------|---|
| PersonID | PersonID of the user. |
| RemedyLoginID | Remedy Login Name. |
| FirstName | First name of the user. |
| LastName | Last name of the user. |
| InternetEmail | Internet Email of user. |
| Status | Status of the user. |
| AccountingNumber | Accounting Number of user. |
| ClientSensitivity | Sensitivity of client. |
| ClientType | Type of Client. |
| Company | Company of user. |
| CorporateID | Corporate ID of user. |
| BusinessPhoneNumber | Business Phone Number of the user. |
| FullTextLicenseType | Full Text License Type of user. |
| JobTitle | Job Title of user. |
| LastModifiedBy | Name of the user who last modified the user attributes. |
| LicenseType | Type of License. |
| Region | Region information. |
| Site | Site information. |
| SiteAddress | Site Address information. |
| SiteGroup | Site Group information. |
| Submitter | Name of the submitter. |
| SupportStaff | If user is part of Support Staff. |
| VIP | If user is VIP. |
| UnrestrictedAccess | If user has unrestricted access. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|--------------------------|-----------------------------------|
| SupportGroupID | Support Group ID |
| Company | Support Company information |
| Description | Group description |
| DisableGroupNotification | If group notification is disabled |
| GroupNotificationEmail | Group notification email id |

Provisioning policy attributes

| Attributes | Description |
|---------------------|--|
| instanceId | Instance id of group |
| LastModifiedBy | Name of the user who last modified the group |
| Status | Status of the group |
| Creator | Creator of the group |
| SupportGroupName | Name of the group |
| SupportGroupRole | Support Group role name |
| SupportOrganization | Support Group organization name |
| UsesOLA | If the group uses OLAs |
| UsesSLA | If the group uses SLAs |
| VendorGroup | If group is a Vendor Group |
| OnCallGroup | If group is a On Call Group |

Provisioning policy attributes

This section lists the different policy attributes of Remedy ITSM Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

| Attributes | Description |
|---------------------|--|
| FirstName | First name of the user. |
| LastName | Last name of the user. |
| ClientType | Type of Client. Following are the allowed values: <ul style="list-style-type: none">■ Office-Based Employee■ Field-Based Employee■ Home-Based Employee■ Contractor■ Customer■ Prospect■ Vendor |
| ClientSensitivity | Sensitivity of Client. Following are the allowed values: <ul style="list-style-type: none">■ Sensitive■ Standard |
| VIP | Following are the allowed values: <ul style="list-style-type: none">■ Yes■ No |
| Company | Company name of the user. |
| BusinessPhoneNumber | Business phone number of the user. |

| Attributes | Description |
|--------------------|--|
| RemedyLoginID | Remedy login Id. |
| 1000005507 | Remedy Password for the login Id. |
| SupportStaff | Following are the allowed values: <ul style="list-style-type: none"> ■ Yes ■ No If value is Yes, AssignmentAvailability attribute needs to be added. Allowed values: Yes or No. |
| UnrestrictedAccess | Following are the allowed values: <ul style="list-style-type: none"> ■ Yes ■ No |

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|---------------------|---|
| SupportCompany | Support Company name of group. |
| SupportOrganization | Support organization of group. |
| SupportGroupName | Support group name. |
| SupportGroupRole | Support Group role name. |
| VendorGroup | Following are the allowed values: <ul style="list-style-type: none"> ■ Yes ■ No |
| OnCallGroup | Following are the allowed values: <ul style="list-style-type: none"> ■ Yes ■ No |

Update policies

The following table lists the attributes for different update policies:

| Attributes | Description |
|--|--------------------------------------|
| Enable/Disable a user | |
| ResetPassword | The new password to be set. |
| Create an ITSM Account and Group Connection | |
| AC_1000000017 | Full name of the user. |
| AC_4 | Remedy Login ID of the user. |
| AC_1000000401 | Support Group Association Role name. |

Note: The connection attributes should have 'AC_' prefixed to the field id of the attribute.

Additional information

This section describes the additional information related to the BMC Remedy ITSM Suite Connector.

Enable/Disable Account

For disabling a user, a password not known to the user should be provided by the administrator. The **Profile Status** attribute of the user will be set to **Obsolete**. All users which have a status other than **Enabled** will be marked as **Disabled** in IdentityIQ.

For enabling a user, a password should be provided by the administrator which can be communicated to the user after successful password change. The **Profile Status** attribute of the user will be set to **Enabled**.

Add Entitlement operation for ITSM

To add a user to a group in BMC Remedy ITSM, there are some mandatory attributes to be provided which are a part of the connection between the user and the group. Hence, for Remedy ITSM, an entitlement will have mandatory attributes which will be a part of the update provisioning policy. All entitlements added will have the same connection attributes.

Troubleshooting

- When an attribute is to be added to the schema, the attribute's ID should be added as an **internalName** of the attribute in the schema.
- When an attribute (which is not present in the schema) is to be added to the provisioning policy, the ID of the attribute should be provided as the **name** of the attribute.
- For connection attributes, ensure that the ID of the attribute is prefixed with **AC_**.
- While creating an ITSM Account having SupportStaff value **Yes**, ensure that the **AssignmentAvailability** attribute is added to the provisioning policy.

For example,

```
<Field displayName="AssignmentAvailability" name="1000000346"
reviewRequired="true" type="string">
  <AllowedValues>
    <String>Yes</String>
    <String>No</String>
  </AllowedValues>
</Field>
```

- For account creation in BMC Remedy ITSM Suite version 7.5.00.001 required mandatory attribute **InternetEmail**.

Chapter 26: SailPoint IdentityIQ Jive Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|-----|
| Overview | 225 |
| Supported features | 225 |
| Pre-requisites | 225 |
| Administrator permission | 226 |
| Configuration parameters | 226 |
| Schema attributes | 226 |
| Account attributes | 226 |
| Group attributes | 227 |
| Provisioning Policy attributes | 228 |
| Create account attributes | 228 |
| Create group attributes | 228 |

Overview

SailPoint IdentityIQ Jive Connector was developed to manage Jive User Accounts and Security Groups (Security Groups may have permissions such as Full Access, Manage Community, Manage System, Moderate Content, Manage Users, Manage Groups) from IdentityIQ.

Supported features

SailPoint IdentityIQ Jive Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Account Refresh
- Create/Delete/Update Account
- Enable/Disable Account
- Create/Delete/Update Group

Pre-requisites

1. Jive connector requires the following libraries in IdentityIQ's class path:
 - httpcore-4.2.1.jar
 - httpclient-4.2.1.jar
 - httpclient-cache-4.2.1.jar
 - commons-logging-1.1.1.jar
 - commons-codec-1.6.jar
 - gson-2.1.jar
2. Jive software should be up and running.
3. Administrator should be configured to have proper access rights for modifying Jive users.

Administrator permission

Administrator should be configured to have proper access rights for modifying Jive users.

Configuration parameters

The following table lists the configuration parameters of Jive Connector:

| Parameters | Description |
|------------|---|
| Jive URL | URL for accessing Jive (For example, http://jive1.jivedev.com) |
| User Name | User Name used for logging into Jive with sufficient rights (administrator credentials.) |
| Password | User's password. |
| Page Size | The maximum size of each data set when querying over large number of objects. Default and max value is 100. |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|----------------|---|
| Id | Identifier (unique within an object type and Jive instance) of the object. |
| username | The login username for the person. This field is required for person creation, but cannot be changed on an update. |
| name | Name components (familyName, givenName and formatted) for the person. |
| location | Geographic location of the person. |
| type | The object type of the object (person). |
| displayName | Formatted full name of this person, suitable for use in UI presentation. If the person has privacy settings that do not allow you to see his or her name, this will be the Jive username instead. |
| published | Date and time when the person was originally created. Visible only to Jive administrators or on your person object. |
| thumbnailurl | URL of the thumbnail (avatar) image for the person. |
| followingCount | Number of people the person is following. |
| followerCount | Number of people following the object. |

| Attributes | Description |
|-----------------|---|
| jive | Jive extensions to OpenSocial person object(it's a complex JSON object). |
| phoneNumbers | Phone numbers belonging to the person, with standard types: fax, home, mobile, other, pager, work. |
| updated | Date and time the person was most recently updated. |
| addresses | Postal addresses belonging to the person, with standard types home, other, pobox, work and value type of address. |
| work_country | Name of the country where person is working. |
| work_locality | Name of the city where person is working. |
| work_street_1 | Street 1 name where person is working. |
| work_street_2 | Street 2 name where person is working. |
| work_postalCode | Zip/Postal code where person is working. |
| work_region | Name of State or Province where person is working. |
| home_country | Name of the country where person lives. |
| home_locality | Name of the city where person lives. |
| home_postalCode | Zip/Postal code where person lives. |
| home_region | Name of state or province where person lives. |
| home_street_1 | Name of street 1 where person lives. |
| home_street_2 | Name of street 2 where person lives. |
| givenName | First name of the person. |
| familyName | Last name of the person. |
| Department | The person's department name. |
| Company | The person's company name. |
| Title | The person's job title name. |
| Biography | The person's biography. |
| Expertise | The person's expertise. |
| Email | This attribute will be used for updating primary email address of a person. |
| emails | Email addresses belonging to this person, with standard types home, other, work and value type of string. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|------------|--|
| id | Identifier (unique within an object type and Jive instance) of the object. |
| name | Name of the security group. |

Provisioning Policy attributes

| Attributes | Description |
|--------------------|---|
| type | Object type of this object ("securityGroup"). |
| description | Description of this security group. |
| administratorCount | Number of administrative members in the security group. |
| memberCount | Number of regular members of the security group. |
| published | Date and time the security group was initially created. |
| updated | Date and time the security group was last updated. |
| federated | Flag indicating that the membership of the group is federated with an external directory service. |
| Admins | List of administrative members in the security group. |
| Members | List of regular members of the security group. |

Provisioning Policy attributes

This section lists the different policy attributes of Jive Connector.

Note: All the attributes marked with * sign are the mandatory attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

| Attributes | Description |
|-------------|---|
| username* | The login username for this person. This field is required for person creation, but cannot be changed on an update. |
| givenName* | First name of the person. |
| familyName* | Last name of the person. |
| Email* | Primary email address of a person. |
| Password* | Password of a person. |

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|-------------|---------------------------|
| name* | Name of the group. |
| description | Text describing the user. |

Chapter 27: SailPoint IdentityIQ Oracle E-Business Suite Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 229 |
| Supported features | 229 |
| Supported Managed Systems | 230 |
| Pre-requisites | 230 |
| Administrator permissions | 230 |
| Configuration parameters | 231 |
| Schema attributes | 231 |
| Account attributes | 231 |
| Group attributes | 232 |
| Provisioning Policy attributes | 232 |
| Create account attributes | 233 |
| Delete account attributes | 233 |
| Create group attributes | 233 |
| Deleting Group (Responsibility) | 234 |
| Deleting entitlement | 234 |
| Troubleshooting | 234 |

Overview

The Oracle E-Business Suite is an integrated suite of development, runtime, and system management tools. It also includes Forms, JDeveloper, Single Sign-On, Oracle Internet Directory, Portal, Discoverer, Web Cache, Integration, Oracle BPEL Process Manager.

SailPoint IdentityIQ Oracle E-Business Suite connector controls the activities related to account/groups by signing in managed system. SailPoint IdentityIQ Oracle E-Business Suite Connector will manage the following entities of Oracle E-Business Suite:

- Account
- Group (Responsibility)

Supported features

SailPoint IdentityIQ Oracle E-Business Suite Connector provides support for the following features:

- Account Aggregation
- Group Aggregation
- Create/Update/Enable/Disable/Refresh Account
- Create/Update/Delete Group
- Set password
- Remove/Request Entitlement

Supported Managed Systems

Following versions of Oracle E-Business Suite are supported by the connector:

- Oracle E-Business Suite 12.0.x
- Oracle E-Business Suite 12.1.x

Pre-requisites

Type 4 thin driver is required for connecting to the Oracle server using JDBC.

By default IdentityIQ is bundling **ojdbc14.jar** file (which works properly with Oracle 10i families). But if the backend Oracle Server is of version greater than 10i, the connector does not function properly with **ojdbc14.jar** file. For more information, see the [“Troubleshooting” on page 234](#) section.

Administrator permissions

- **For GET operation (Aggregation):** Administrator should have the SELECT permission on the following tables:

| Tables | SELECT permission |
|-------------|--|
| Account | FND_USER PER_PEOPLE_F RA_CUSTOMERS |
| Group | FND_RESPONSIBILITY_VL FND_APPLICATION_VL FND_DATA_GROUPS |
| Entitlement | FND_USER_RESP_GROUPS_ALL |

- **For SET operation (Create/Update/Delete):** Administrator should have **execute** permission on the following stored procedure:

| Entity | Operation | Stored procedure with PKG |
|------------------------|-----------------|-----------------------------------|
| Account | Create | FND_USER_PKG.CreateUser |
| | Update | FND_USER_PKG.UpdateUser |
| | Delete | FND_USER_PKG.DisableUser |
| | Enable | FND_USER_PKG.UpdateUser |
| | Disable | FND_USER_PKG.DisableUser |
| | Change Password | fnd_web_sec.change_password |
| Group (Responsibility) | Create | FND_RESPONSIBILITY_PKG.INSERT_ROW |
| | Update | FND_RESPONSIBILITY_PKG.UPDATE_ROW |
| | Delete | FND_RESPONSIBILITY_PKG.UPDATE_ROW |
| Entitlement | Request | FND_USER_PKG.AddResp |
| | Remove | FND_USER_PKG.DelResp |

Configuration parameters

The following table lists the configuration parameters of Oracle E-Business Suite Connector:

Note: Attributes marked with * sign are the mandatory attributes.

| Attributes | Type |
|----------------------|---|
| Connection User* | The Oracle EBS Login name through which we want to connect Oracle EBS. For example, apps . |
| Connection Password* | The authentication details of login. |
| Database URL* | The URL for server which directly interact with the Managed system. For example, jdbc\:oracle\:thin\:@localhost\:1521\:DATABASENAME jdbc\:oracle\:thin\:@localhost\:1521\:ABC |
| JDBC Driver* | It is the name of the Driver class supported by JDBC Type 4. For example, oracle.jdbc.driver.OracleDriver |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|---------------------|--|
| USER_NAME | Application username (what a user types in at the Oracle Applications sign-on screen). |
| USER_ID | Application user identifier. |
| START_DATE | The date the user name becomes active. |
| END_DATE | The date the user name becomes inactive. |
| DESCRIPTION | Description. |
| PASSWORD_DATE | The date the current password was set. |
| PASSWORD_EXPR | The number of accesses left for the password. |
| PASSWORD_NO_OF_DAYS | The number of accesses allowed for the password. |
| EMAIL_ADDRESS | The electronic mail address for the user. |
| FAX | The fax number for the user. |
| EMPLOYEE_ID | Identifier of employee to whom the application username is assigned. |

Provisioning Policy attributes

| Attributes | Description |
|-----------------|--|
| EMPLOYEE_NUMBER | Unique number of the employee. |
| FULL_NAME | Full name of the user. |
| CUSTOMER_ID | Customer contact identifier. If the AOL user is a customer contact, this value is a foreign key to the corresponding customer contact. |
| CUSTOMER_NAME | Customer name. |
| ROLES | Role created for every responsibility. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|-------------------------|---|
| RESPONSIBILITY_ID | Responsibility identifier. |
| RESPONSIBILITY_NAME | Name of the responsibility. |
| RESPONSIBILITY_KEY | Internal developer name for responsibility. |
| START_DATE | The date the responsibility becomes active. |
| END_DATE | The date the responsibility expires. |
| DESCRIPTION | Description |
| STATUS | Shows status of the responsibility. |
| VERSION | Version |
| WEB_HOST_NAME | IP address or alias of the computer where the Webserver is running. Defaults to the last agent. |
| WEB_AGENT_NAME | Name of Oracle Web Agent. Defaults to the last agent. |
| DATA_GROUP_APPL_NAME | Name of the data group application. |
| REQUEST_GROUP_APPL_NAME | Request Group Application name. |
| DATA_GROUP_ID | Identifier of data group. |
| DATA_GROUP_NAME | Name of the Data Group. |
| MENU_NAME | Name of the menu. |
| REQUEST_GROUP_NAME | Request group name. |

Provisioning Policy attributes

This section lists the single provisioning policy attributes of Oracle E-Business Suite Connector that allows you to select the type of user/group to create.

Note: Attributes marked with * sign are the mandatory attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

| Attributes | Description |
|--------------------------|---|
| Name* | Name of the login user. |
| Password* | Password of the login user. |
| Description | Description. |
| Start Date* | The date the responsibility becomes active. |
| End Date | The date the responsibility expires. |
| Password Expiration Type | Type of the password to expire. |
| Password Expiration Type | Days after which user password will expire. |

Delete account attributes

In delete operation, Oracle E-Business Suite sets the value of **END DATE** parameter of user to current system date, this operation is nothing but the **disable** operation.

When the delete operation is performed as a result Oracle E-Business Suite connector would display an error. In such case administrator should use the disable user operation.

Create group attributes

The following table lists the provisioning policy attributes for Create Group (Responsibility):

| Attributes | Description |
|--------------------------------|---|
| Responsibility Name* | Name of the responsibility. |
| Application Name* | Name of the application. |
| Description | Description |
| Responsibility Key* | Internal developer name for responsibility. |
| Start Date* | The date the responsibility becomes active. |
| End Date | The date the responsibility expires. |
| Responsibility Version* | Responsibility version. |
| Data Group Name* | Name of the data group. |
| Data Group Application Name* | Name of the data group application. |
| Menu Name* | Name of the menu. |
| Request Group Name | Request group name. |
| Request Group Application Name | Request group application name. |

Deleting Group (Responsibility)

In delete operation, Oracle E-Business Suite sets the value of END DATE parameter of group to current system date. This operation is the **disable** operation.

When the delete operation is performed as a result this will display an error and instruct the administrator to configure the **ORAAPPL_DEACTIVATE_RESP** parameter and set the value to **Y**. This will disable the group.

Deleting entitlement

In delete operation, Oracle E-Business Suite sets the value of **END DATE** parameter of entitlement to current system date. This operation is the **disable** operation.

When the delete operation is performed as a result this will display an error and instruct the administrator to configure the **ORAAPPL_ACTUAL_DEL_ENTITLEMENT** parameter and it should be set to **Y** then Entitlement can be disabled.

If **ORAAPPL_ACTUAL_DEL_ENTITLEMENT** is set to **Y** then entitlement will not be displayed after aggregation.

Troubleshooting

1 - If the version of Oracle Server is greater than 10i, the connector does not work properly with ojdbc14.jar file.

For example, for Oracle version 11g the ojdbc jar file which is required for proper functioning of the connectors is **ojdbc6.jar** file.

Resolution: Perform the following to obtain the required version of the ojdbc jar file:

1. Download the suitable ojdbc jar file for oracle from the <http://www.oracle.com> site to a temporary directory.
2. Turn off the webserver.
3. Rename the already bundled **ojdbc14.jar** file to **ojdbc14.jar_old** in the **..\identityiq\WEB-INF\lib** directory.
4. Copy the latest downloaded jar file in step 1 from the temporary directory to the **..\identityiq\WEB-INF\lib** directory.
5. Restart the webserver.

Chapter 28: SailPoint IdentityIQ Rally Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 235 |
| Supported features | 235 |
| Pre-requisites | 235 |
| Administrator permission | 235 |
| Configuration parameters | 235 |
| Schema attributes | 236 |
| Provisioning Policy attributes | 236 |

Overview

SailPoint IdentityIQ Rally Connector can be used to manage Rally User accounts and to revoke their permission (workspace permission and project permission) from IdentityIQ.

Supported features

SailPoint IdentityIQ Rally Connector provides support for the following functionalities:

- Account Aggregation
- Account Refresh
- Create/Update/Delete Account
- Enable/Disable Account
- Removal of direct permissions (User Permissions)

Pre-requisites

Rally software should be up and running.

Administrator permission

Workspace administrator or subscription administrator should be configured to have proper access rights for modifying Rally users.

Configuration parameters

We make use of Rally REST based web services to communicate with Rally System. Below information is required for connecting to Rally System.

| Parameters | Description |
|------------|--|
| Rally URL | URL for accessing Rally (For example, https://rally1.rallydev.com). |

Schema attributes

| Parameters | Description |
|------------|---|
| User Name | User Name used for logging into Rally with sufficient rights (workspace administrator with rights to create users). |
| Password | User's Password. |
| Page Size | The number of objects to fetch in a single page when iterating over large data sets. The default size is 200. The value should be valid number. |

Schema attributes

The following table lists the account attributes:

| Attributes | Description |
|------------------|--|
| ObjectID | User's unique ID. |
| CreationDate | Date when the user was created. |
| DisplayName | Preferred name to be used for the person throughout the application. |
| EmailAddress | Email address of the user. Administrative messages and email notifications will be sent to this email address. |
| FirstName | First name of the user. |
| LastName | Last name of the user. |
| MiddleName | Middle name of the user. |
| Disabled | The accounts inactive status. This field is only available to subscription or workspace administrators. Inactive users are unable to log into Rally. |
| Role | Role of the user. |
| ShortDisplayName | User's short name. |
| SubsrcitionAdmin | Subscription administrator. |
| UserPermissions | The user's specific permissions for each workspace in the subscription. |
| TeamMemberships | Information about the user's membership to projects. |

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Accounts:

| Attributes | Description |
|--------------|---|
| DisplayName | The preferred name to be used for this person throughout the application. |
| EmailAddress | An email address used for administrative messages and email notifications. |
| UserName | An email address . Rally recommends using a valid email address for each of your users. |

Provisioning Policy attributes

| Attributes | Description |
|------------|-------------------------|
| FirstName | First name of the user. |
| LastName | Last name of the user. |

Provisioning Policy attributes

Chapter 29: SailPoint IdentityIQ BMC Remedy Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 239 |
| Supported features | 239 |
| Supported Managed Systems | 239 |
| Pre-requisites | 240 |
| Administrator permission | 240 |
| Configuration parameters | 240 |
| Schema attributes | 240 |
| Account attributes | 240 |
| Group attributes | 241 |
| Provisioning policy attributes | 241 |
| Create account attributes | 241 |
| Create group attributes | 242 |
| Update policies | 242 |
| Additional information | 242 |
| Enable/Disable Account | 242 |
| Troubleshooting | 243 |

Overview

SailPoint IdentityIQ BMC Remedy Connector was developed to manage the accounts and groups contained in BMC Remedy Action Request System.

Supported features

SailPoint IdentityIQ BMC Remedy Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Account Refresh
- Create/Delete/Update Account
- Enable/Disable Account
 - For more information, see [“Enable/Disable Account”](#) on page 242.
- Add/Remove Entitlements
- Create/Update/Delete Account Group
- Reset/Change Password
- Pass through Authentication

Supported Managed Systems

- BMC Remedy Action Request System Server version 8.1
- BMC Remedy Action Request System Server version 8.0
- BMC Remedy Action Request System Server version 7.6.04
- BMC Remedy Action Request System Server version 7.5.01

Pre-requisites

1. You must copy the **arapi<v>.jar** file from the location where the server is installed (*installFolder*\BMC Software\ARSystem\midtier\WEB-INF\lib) to the lib folder of IdentityIQ installation (\webapps\identityiq\WEB-INF\lib).
2. Add the location of **arapi<v>.jar** file to the CLASSPATH system variable (\webapps\identityiq\WEB-INF\lib\arapi<v>.jar) of the computer where IdentityIQ installed.
3. Provide the appropriate read and write permissions to the Administrator to perform the user and group provisioning operations from IdentityIQ.

Administrator permission

The Application User should be a member of the **Administrator** group.

Configuration parameters

The following table lists the configuration parameters of BMC Remedy Connector:

| Parameters | Description |
|----------------------------------|---|
| Remedy Server name or IP Address | IP address of the computer on which the Remedy server is installed. |
| Administrator Name | Name of the Remedy administrator. |
| Administrator Password | Password of the administrator. |
| Server Port | Remedy Server port number. |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|----------------------------|--|
| RequestID | RequestID of the user. |
| LoginName | Remedy login name. |
| ForcePasswordChangeOnLogin | Set to Yes if the user should be asked to change his password on next login else to No. |
| FullName | Full name of the user. |
| Status | Status of the user. |
| AccountDisabledDate | Account disabled date of user. |

| Attributes | Description |
|--|--|
| ApplicationLicense | Application license of user. |
| AppliedDaysAfterExpirationUntilDisablement | Applied days after expiration until disablement of user. |
| AppliedNewUserMustChangePassword | Is set to Yes if the new user must change password. |
| AppliedNo.DaysbeforeExpiration | The number of days before expiration. |
| AppliedNumberofWarningDays | The number of warning days. |
| AppliedPasswordEnforcementEnabled | Is set to Yes if password enforcement is enabled. |
| Creator | Creator of the user. |
| LastModifiedBy | Name of the user who last modified the user. |
| LicenseType | License Type of user. |
| UniquelIdentifier | Unique Identifier of the user. |
| Groups | Groups connected to the user. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|-------------------|---|
| RequestID | RequestID of group. |
| Comments | Comments about group. |
| GroupCategory | Category of group. |
| GroupID | ID of group. |
| GroupName | Name of the group. |
| GroupType | Type of group. |
| LastModifiedBy | Name of the user who last modified the group. |
| LongGroupName | Long name of the group. |
| ParentGroup | Parent group of the group. |
| Status | Status of the group. |
| UniquelIdentifier | Unique Identifier of group. |

Provisioning policy attributes

This section lists the different policy attributes of BMC Remedy Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

Additional information

| Attributes | Description |
|--------------------------------|---|
| Login Name | Remedy login name of the user. |
| Full Name | Full name of the user. |
| Force Password Change On Login | Is set to Yes if the user should be asked to change his password on next login. |
| License Type | License Type of the user. |
| Password | Password of the user. |

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|-----------------|---|
| Group Name | Name of the group to be created. |
| Group ID | ID of the group. It should be a numeric value. |
| Group Type | Type of the group. |
| Long Group Name | Long name of the group. |
| Group Category | Category of the group. If the category of the group is Computed , ComputedGroupDefinition needs to be added in the provisioning policy. |

Update policies

The following table lists the attributes for enable/disable a user:

| Attributes | Description |
|---------------|-----------------------------|
| ResetPassword | The new password to be set. |

Additional information

This section describes the additional information related to the BMC Remedy Connector.

Enable/Disable Account

For disabling a user, a password not known to the user should be provided by the administrator. The **Status** attribute of the user will be set to **Disabled**. All users which have a status other than **Current** will be marked as **Disabled** in IdentityIQ.

For enabling a user, a password should be provided by the administrator which can be communicated to the user after successful password change. The **Status** attribute of the user will be set to **Current**.

Troubleshooting

- When an attribute is to be added to the schema, the attribute's ID should be added as an **internalName** of the attribute in the schema.
- When an attribute (which is not present in the schema) is to be added to the provisioning policy, the ID of the attribute should be provided as the **name** of the attribute.
- While creating a Remedy Group having GroupType value **Computed**, ensure that the **ComputedGroupDefinition** attribute is added to the provisioning policy.
For example, `<Field displayName="ComputedGroupDefinition" name="121" type="string"/>`

Chapter 30: SailPoint IdentityIQ RSA Ace Server Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|-----|
| Overview | 245 |
| Supported features | 245 |
| Supported Managed Systems | 245 |
| Pre-requisites | 246 |
| Administrator permissions | 247 |
| Configuration parameters | 247 |
| Schema attributes | 248 |
| Account attributes | 248 |
| Group attributes | 248 |
| Provisioning Policy attributes | 249 |

Overview

SailPoint IdentityIQ RSA Ace Server Connector manages the Users, Groups and Access Tokens in the RSA Ace Server. The connector manages the following entities:

- Users
- Groups
- Administrative Roles
- Secure ID tokens

The RSA groups are considered as the account-group with support for provisioning (Create, Update, and Delete) while Administrative Roles are additional entitlements which can only be assigned or removed.

Supported features

SailPoint IdentityIQ RSA Ace Server-Direct Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Account Refresh
- Add/Remove entitlements
- Create/Update/Delete Account
- Create/Update/Delete Account-Group
- Enable/Disable/Unlock Account
- Reset/Change Password
- Pass through Authentication
- Aggregation/Revocation of Direct Permissions for Secure ID tokens

Supported Managed Systems

SailPoint IdentityIQ RSA Ace Server Connector supports RSA Authentication Manager version 7.1 SP4.

Pre-requisites

- **Configuration for SSL version 3 support**

The Authentication Manager only accepts SSL version 3 (or higher) requests. For any other versions of SSL an error message similar to the following is displayed:

```
[java] ERROR: javax.net.ssl.SSLException: Received fatal alert: bad_record_mac
[java] com.rsa.common.SystemException: javax.net.ssl.SSLException: Received fatal
alert: bad_record_mac
[java] at
com.rsa.webservice.SOAPCommandTarget.remoteMethod(SOAPCommandTarget.java:198)
[java] at
com.rsa.webservice.SOAPCommandTarget.executeCommand(SOAPCommandTarget.java:136)
[java] at com.rsa.command.TargetableCommand.execute(TargetableCommand.java:241)
```

To allow communication over SSL version 3, add the following Java options to the application server:

```
-Daxis.socketSecureFactory=com.rsa.webservice.net.SSLv3JSSESocketFactory
```

- **Configuring the Trust Store**

Server root certificate should be imported into the keystore for the remote API calls. Ensure to add the following Java option to the application server for SSL SOAP connections:

```
-Djavax.net.ssl.trustStore2 = <Path of the of the imported root certificate>
```

- **Importing the Server Root Certificate (Java)**

When RSA Ace Server is installed, the system creates a self-signed root certificate and stores it in **RSA_AM_HOME/server/security/server_name.jks** directory. This certificate must be exported from the server, and import it into the keystore for remote API clients. Use the Java keytool, as described in the following sections to export and import the certificate into Java clients.

- **To export the server root certificate:**

- a. Change directories to **RSA_AM_HOME/appserver/** and enter the following:

```
jdk/jre/bin/keytool -export -keystore
```

```
RSA_AM_HOME/server/security/server_name.jks -file am_root.cer -alias rsa_am_ca
```

- b. At the prompt for **keystore_password**, press **Enter** without the password.

Note: Ignore the warning message that appears as the server root certificate will still be exported.

- c. The Java keytool outputs the certificate file to the **RSA_AM_HOME/appserver/** directory.

- **To import the server root certificate (Java):**

Note: You must provide your cacerts keystore password to import the server root certificate. The Java default is "changeit".

- a. Locate the server root certificate file that you exported from Authentication Manager, and copy it to the target host.
- b. Import the certificate to the local cacerts keystore. Change directories to **JAVA_HOME/jre/bin**, and enter the following:

```
keytool -import -keystore SDK_HOME/lib/java/trust.jks -storepass
```

```
cacerts_keystore_password -file am_root.cer -alias rsa_am_ca -trustcacerts
```

- c. The Java keytool displays a confirmation that the certificate was added to the keystore.

Administrator permissions

The RSA Ace Server Connector administrator must have enough rights to execute the requested operation. To assign the rights to the administrator:

- Assign the default administrative roles present on the RSA. For most of the operations **Auth Mgr Realm Admin** administrative role should be assigned.
- Create new administrative roles with relevant permissions and assigning it a scope and then assign it to the administrator. The scope of an administrative role determines in what security domains an administrator may manage objects and from what identity sources an administrator may manage users. Below are the permissions which can be assigned to the administrative role.
 - **All** grants an administrator permission to perform any administrative action on the object.
 - **Delete** grants an administrator permission to delete an object.
 - **Add** grants an administrator permission to add an object.
 - **Edit** grants an administrator permission to view and edit an object, but not the ability to add or delete.
 - **View** grants an administrator permission to view an object, but not to add, edit, or delete.

Configuration parameters

The following table lists the configuration parameters of RSA Ace Server Connector:

| Parameters | Description |
|-------------------------|---|
| Host | The RSA Ace Server to connect to. |
| Port | The port to use to connect to RSA Ace Server. Default: 7002. |
| Administrator | The account that has permission to connect to the RSA Ace Server resource remotely. This account should have permission to manage this resource. |
| Password | Password of the Administrator account. |
| Command Client User | The command client user name. On installation of RSA Ace Server, the system creates a command client user name and password for secure connections to the command server. This user name and password are randomly generated on creation, and are unique to each deployment. |
| Command Client Password | Command Client Password corresponding to the Command Client User. |
| Realm | Name of the Realm to manage. |
| Identity Source | Identity Source name linked to the Realm. |
| Security Domain | Name of the security domain to manage. |
| Search Subdomain | Whether or not to manage the subdomain, when the parent security domain is specified for Security Domain field. |
| Page Size | Limit to fetch number of accounts or groups per iteration through RSA Ace Server. Default: 500. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

| Attributes | Data type | Description |
|---------------------|--------------------|--|
| Guid | String | Guid of the entity (Native Identity). |
| userID | String | Unique name by which the entity is known by. (Display Attribute). |
| firstName | String | First name of the entity. |
| middleName | String | Middle name of the entity. |
| lastName | String | Last name for which the entity is known by. |
| Notes | String | Notes or description for the entity. |
| Email | String | Email of the entity. |
| certificateDN | String | Certificate DN of the entity. |
| securityDomain | String | Security Domain Name to which entity belongs. |
| identitySource | String | Identity Source Name to which entity belongs. |
| lastModifiedBy | String | Administrator or user who modified the entity last time. |
| Groups | Multivalued String | Groups Membership (marked as "groupAttribute"). |
| Roles | Multivalued String | Administrative roles assigned to the entity. |
| lastModifiedOn | String | Last time when the entity was modified. |
| accountStartDate | String | Time when the entity was created. |
| accountExpireDate | String | Time when the entity will get expired. |
| lastLogin | String | Last time when the entity was logged in. |
| forceChangePassword | Boolean | Whether or not user need to change the password during next logon. |

Group attributes

The following table lists the group attributes:

| Attributes | Data type | Description |
|----------------|-----------|---|
| guid | String | Guid of the entity (Native identity). |
| groupName | String | Name by which the entity is known by (Display Attribute). |
| Notes | String | Notes or description for the entity. |
| securityDomain | String | Security Domain Name to which entity belongs. |
| identitySource | String | Identity Source Name to which entity belongs. |

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Account of RSA Ace Server Connector.

Note: The attributes marked with * sign are the required attributes.

| Attributes | Description |
|---------------------|--|
| userID* | Unique name by which the entity is known by. |
| password* | Password for RSA user. |
| lastName* | Last Name of the user. |
| firstName | First name of the user. |
| email | Email of the user. |
| forceChangePassword | Whether or not user need to change the password during next logon. |
| nextAvailableToken | Select to assign the next available SecureID token to the user. |

Provisioning Policy attributes

Chapter 31: SailPoint IdentityIQ Salesforce/Remedyforce Connector

The following topics are discussed in this chapter:

| | |
|---|-----|
| Overview | 251 |
| Supported features | 251 |
| Administrator permissions | 252 |
| Configuration parameters | 252 |
| Schema attributes | 253 |
| Account attributes | 253 |
| Additional account attributes for Remedyforce connector | 255 |
| Profile attributes | 255 |
| Provisioning Policy attributes | 256 |

Overview

The Salesforce/Remedyforce connector provides support for reading and provisioning of Salesforce/Remedyforce accounts, profiles as account groups and implement the **sailpoint.connector** interface.

This connector is written using the **partner.wsdl** and underlying soap interface. The connector uses SOAP stub generated from a wsdl that was available at the time of development. The stubs are generated using axis 1.2. We do not have to generate the stubs once already done. Partner API is easy to use and we can add custom attributes in the schema without generating the stubs. Partner API is generic and have the same java implementation for Salesforce and RemedyForce connectors.

The API is fairly rich for SOAP based API and has the concept of login which requires us to login just once for each operation. It has formal models around the user and profile objects and they are generated as part of the stubs. Earlier Salesforce Connector internally used Enterprise WSDL which was complex to use (regenerating STUB classes by customer) and not much flexible on custom attributes. With Partner WSDL approach, those limitations will be removed. RemedyForce extends account schema of salesforce to accommodate extra attributes related to BMC Remedy systems.

Supported features

The Salesforce/Remedyforce connector provides support for the following functionalities:

- Account Aggregation
- Group Aggregation

Note: Aggregates salesforce/remedyforce.com profiles as account groups/managed attributes.

- Create/Update Account

Note: The delete functionality will disable the user account as Salesforce/RemedyForce.com does not support delete.

- Enable/Disable User (Only for Salesforce)
- Get Direct permissions for groups

Configuration parameters

Note: Group Aggregation will aggregate direct permissions for groups.

- Reset/Change Password

Note: Reset Password Operation will not set password provided from IdentityIQ for the user but it will send Email Notification with temporary password to the user.
Change Password does not require current password.

- Add Entitlement for user (Only Profiles and User Roles can be added as user entitlements)

Administrator permissions

For user provisioning through IdentityIQ, required that the administrator have the appropriate rights on the Salesforce Account.

The System Administrator Profile can configure and customize the application.

- Has access to all functionality that does not require an additional license. Can create, edit, and delete custom profiles. Can reset password of multiple users.
- Can add Multiple Users.
- Has access to all Users, Profile Permissions.
- Enable /Disable Users

Profile Access to User

A user Profile determines what a user can do in the system. By default, the System Administrator Profile can do the most; the Read Only Profile can do the least. For most users, the Standard User Profile is a good choice: it lets people create and edit most records, as well as access and run reports.

(The following assumes that you are a System Administrator for your organization's instance of Salesforce.com.)

Users, Roles and Profiles are all configured within the Setup area. To access these settings when logged in to Salesforce, click on your name in the upper right corner, then choose Setup from the drop-down menu. The Users, Roles and Profiles settings are all available under Manage Users in the lower left Administration Setup menu.

User Licenses create access

Most of your users will need a standard Salesforce user license. This license gives the user full access to Salesforce's CRM features and applications, including Chatter. Other user license options limit user access.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Salesforce/Remedyforce connector uses the following connection parameters:

Table 32—Configuration parameters

| Parameters | Description |
|-------------------------------|--|
| Salesforce URL* | Enter the fully qualified url to the root of the salesforce/remedyforce server. For example, <code>http://login.salesforce.com/services/Soap/u/26.0/</code> Note: To figure out the url of your site, login to salesforce/remedyforce.com. Click Develop under the Application heading toward bottom. Next, click API > Generate Partner WSDL, and click Generate. The URL is located under the SalesforceService service name. |
| Username* | Specifies the name of the user id that should be used when connecting to Salesforce/Remedyforce.com. |
| Password* | Defines the password which is used for logging in the managed System. |
| Manage Active Accounts | Retrieves the active accounts during account aggregation. Otherwise it retrieves all the accounts which are enabled/disabled while account aggregation. |
| Search Query for User/Profile | The search query for User/Profile is customizable from customer perspective. If the user wants to retrieve specific attributes which are defined in schema, it can be retrieved for all the users with specific criteria while account/group aggregation by executing the following query: <ul style="list-style-type: none"> ■ User: select Id from User where IsActive = true ■ Profile: select Id from Profile where Name = "xyz" |

Note: In the above table all the attributes marked with * sign are mandatory attributes.

Schema attributes

This section provides the different attributes of the Account attributes and Profile attributes for Salesforce/Remedyforce connector.

Account attributes

The Salesforce/Remedyforce connector returns several attributes falling into two categories. The first are general attributes: name, city, state, and so on. Additionally, there are entitlement attributes that specifies user level access granted to Salesforce/Remedyforce:

| Attributes | Description |
|------------|--|
| UserName | By default, this attribute is the connectors default nativeIdentity AND display name attributes. It's typically in an email type format. For example, <code>denise.hunt@demoexample.com</code> |
| Id | Internal salesforce/remedyforce.com id, not friendly looks like "005A00000014ySyIXX" |
| Name | User's fullname |
| FirstName | User's firstname |

Schema attributes

| Attributes | Description |
|------------------------------------|--|
| LastName | User's lastname |
| Alias | User's assigned alias |
| City | User's city |
| CommunityNickname | DisplayNames for user's online communities |
| CallCenterId | User's call center |
| CompanyName | User's company name |
| Country | User's country |
| Department | User's department |
| Email | User's Email address |
| Division | User's division |
| EmployeeNumber | User's employee number |
| Extension | User's telephone extension |
| Fax | User's fax number |
| IsActive | Flag that indicates if the user is active in sf. False would indicate disabled. |
| EmailEncoding | Encoding that should be used during email communications |
| ProfileId | ID of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected. |
| ProfileName | Name of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected. |
| UserRoleId | User Role's Id. |
| UserNameld | User Role's name. |
| PublicGroups | Publicgroups are the entitlements for user in IdentityIQ. |
| UserPermissionsMarketingUser | Maps to the Marketing User Flag. |
| UserPermissionsMobileUser | Maps to the Mobile User Fag. |
| UserPermissionsOfflineUser | Maps to the Offline user Flag. |
| Phone | User's phone number. |
| ReceivesAdminInfoEmails | Receive the salesforce/remedyforce.com administrator newsletter. |
| UserType | Type of the user. |
| UserPermissionsCallCenterAutoLogin | Maps to Force.com Flow user. |

| Attributes | Description |
|------------------------------|--|
| UserPermissionsAvantgoUser | Maps to Avantgo User. |
| UserPermissionsSFContentUser | Maps to Sales Anywhere User. |
| ReceivesInfoEmails | Receive the salesforce/remedyforce.com newsletter. |

Additional account attributes for Remedyforce connector

In addition to the above account attributes, following are the additional custom attributes which are required for configuration to connect to Remedyforce connector:

| Attributes | Description |
|---|------------------------------------|
| BMCServiceDesk__IsStaffUser__c | Maps to BMC ServiceDesk Staff |
| BMCServiceDesk__Remedyforce_Knowledge_User__c | Maps to Remedyforce Knowledge User |
| BMCServiceDesk__Account_Name__c | Maps to Account Name |
| BMCServiceDesk__remarks__c | Maps to Remarks |
| BMCServiceDesk__IsOutOfOffice__c | Maps to Out of Office |
| BMCServiceDesk__ContactId__c | Maps to Contact Id |
| BMCServiceDesk__Account_ID__c | Maps to Account ID |
| BMCServiceDesk__FPLoginID__c | Maps to FootPrints Login ID |
| BMCServiceDesk__Room__c | Maps to Room attribute |

Profile attributes

Profiles are aggregated during account group aggregation, below are the attributes returned by the group aggregation process.

| Attributes | Description |
|-------------|--|
| Id | The internal id for this group. For example, 00eA00000000oP6IAK. |
| Name | The friendly name assigned to the profile. For example, Force.com - Free User, it also has to be unique so is used as the identity and display attribute by default. |
| UserType | This is the type of profile even though the attribute name would indicate a user. |
| Description | Description for the profiles. |

DirectPermissions: The connector reads the permissions assigned to a profile using the salesforce/remedyforce.com api. To get the permissions, the connector queries the service to describe the profile object. In the returned attribute all of the permissions contained by a group are prefixed with **Permissions**, and camel cases the permission such that right and target are separated by camel case convention. For example, **PermissionsEditTask** or **PermissionsTransferAnyEntity**. We break these down into a Permission attribute per prefixed-attribute.

Provisioning Policy attributes

IdentityIQ has a default Provisioning Policy defined which allows for the creation of accounts. The provisioning policy can be edited to fit specific customer environments.

Most of the fields on the Salesforce/Remedyforce connector default provisioning policy are generated and all fields are marked review required. The provisioning policy attributes must be customized based on specific customer requirements.

| Attributes | Description |
|---------------------------|--|
| Create User Policy | |
| Alias | 8 character alias, which is required. By default, it generates a value based on lastname and firstname in the field's inline script. It takes first 7 chars from last name and prefixes it with the the first character of the first name. |
| IsActive | Defaults to true. |
| Username | Defaults to the identity's email address. |
| Email | Defaults to the identity's email address. |
| FirstName | Defaults to the identity's first name. |
| LastName | Defaults to the identity's last name |
| CommunityNickname | Defaults to identity's full name. |
| TimeZoneSidKey | Defaults to America/Los_Angeles. The timezone of the user, it uses a display name defined by sales force. Only a few timezones are defined in the policy drop down and this will need to be customized for each customer. |
| LocaleSidKey | Defaults to UTF-8. This is the user's locale. |
| Email EncodingKey | Defaults to UTF-8 and there are several selections to choose from in from the web interface. They can be customized by the customer. |
| LanguageLocaleKey | Defaults to en_US. There are several selections to choose from in from the web interface. They can be customized by the customer. |

Chapter 32: SailPoint IdentityIQ SAP Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 257 |
| Supported features | 257 |
| Supported Managed Systems | 258 |
| Pre-requisites | 258 |
| Administrator permissions | 258 |
| Configuration parameters | 258 |
| Schema attributes | 259 |
| Account attributes | 259 |
| Group attributes | 262 |
| Schema extension and custom attributes | 263 |
| Provisioning Policy attributes | 263 |
| Create account attributes | 263 |
| Additional information | 263 |
| Entitlement validity period | 264 |
| CUA support | 264 |
| Troubleshooting | 264 |

Overview

SAP Enterprise Resource Planning software solution is an integrated software solution that incorporates the key business functions of the organization.

The SAP Connector returns all the users and roles of the SAP system as well as provisions users and their roles and/or profiles to the SAP system.

SailPoint IdentityIQ SAP Connector was developed to enhance the connector to support provisioning capabilities to a standalone SAP system as well as to a SAP Central User Administration (CUA) system.

Supported features

SailPoint IdentityIQ SAP Connector provides support for the following features:

- Password Reset
- Create Account
- Delete Account
- Enable/Disable/ Account
- Request/Remove Entitlement (for standalone and CUA SAP System)
- Pass-through Authentication

Supported Managed Systems

Following versions of SAP system are supported by the SAP connector:

- SAP NetWeaver 7.0
- SAP NetWeaver 7.1
- SAP NetWeaver 7.2
- SAP NetWeaver 7.3
- SAP NetWeaver 7.31

Pre-requisites

SAP JCO version 3.0.x libraries, along with **sapjco3.dll** (on Microsoft Windows) or **libsapjco3.so** (on UNIX), must be present in the <SPHOME>\WEB-INF\lib directory on the IdentityIQ host. The JCO libraries (JCO Release 3.0.x) must be downloaded from the SAP website by navigating to the customer service marketplace and download the Java Connector (under Tools and Services).

Administrator permissions

The role assigned to the SAP Administrative user must have the following Authorization Objects assigned to it along with the respective activities (permissions):

- S_RFC (All Activities)
- S_USER_AGR (Activities: 02, 03, 22, 36, 78)
- S_USER_GRP (Activities: 01, 02, 03, 05, 06, 22, 78)
- S_USER_PRO (Activities: 01, 02, 03, 06, 07, 22)
- (Additionally for SAP CUA System) S_USER_SAS (Activities: 01, 06, 22)
- (Additionally for SAP CUA System) S_USER_SYS (Activities: 03, 59, 68, 78)

Configuration parameters

The following table lists the configuration parameters of SAP Connector:

| Parameters | Description |
|------------------|--|
| SAP Host* | Host on which the SAP Server is running |
| System Number* | 2-digit SAP system number (Default: 00) |
| Client Number* | 3-digit SAP client number (Default: 001) |
| Client Language* | 2-letter SAP client language (Default: EN) |
| Username* | SAP Administrative user |
| Password* | SAP Administrative user password |
| CUA system | For CUA system detection |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|----------------------------------|-------------------------------------|
| Academic Title (Address) | Academic title of the user. |
| Academic Title 2 (Address) | 2nd Academic title of the user. |
| Addr Number (Address) | Address number of the user. |
| Alias (Logon Data) | Alias name. |
| Birth Name (Address) | Name at birth. |
| Building (Address) | Name of the building. |
| Building 2 (Address) | Name 2 of the building. |
| Building Long (Address) | Long name of the building. |
| Care of (Address) | Care of name. |
| Check Status (Address) | Check status for the user. |
| City (Address) | Name of the city. |
| City Number (Address) | Number of the city. |
| Code (Address) | Signature initials |
| Communication Language (Address) | Communication language of the user. |
| Communication type (Address) | Communication method for the user. |
| Company (Address) | Name of the company. |
| Company Address (Address) | Address of the company. |
| Company Address 2 (Address) | Address 2 of the company. |
| Company Address 3 (Address) | Address 3 of the company. |
| Company Address 4 (Address) | Address 4 of the company. |
| Country (Address) | Name of the country. |
| Country ISO (Address) | ISO name of the country. |
| Delivery District (Address) | Delivery district name. |
| Department (Address) | Department name. |
| District (Address) | District name. |
| District Number (Address) | District number for the user. |
| E-Mail (Address) | E-mail address. |

Schema attributes

| Attributes | Description |
|-----------------------------|---|
| E-Mail List (Address) | E-mail address list. |
| Employee Number (Address) | Employee number of the user. |
| Fax (Address) | Fax number. |
| Fax Extension (Address) | Fax extension number |
| Fax List (Address) | Fax number list |
| First name (Address) | First name of the user |
| Floor (Address) | Floor number |
| Floor 2 (Address) | Floor 2 number |
| Format (Address) | Format name |
| Full Name (Address) | Full name of the user |
| Full Name 2 (Address) | Full name 2 of the user |
| Function (Address) | Function of the user |
| House Number 2 (Address) | House number 2 of the user |
| House Number (Address) | House number of the user |
| House Number 3 (Address) | House number 3 of the user |
| Inhouse ML (Address) | Inhouse mail of the user |
| Initials (Address) | Initials of the user |
| Language CR P (Address) | CR P language of the user |
| Language ISO (Address) | ISO language of the user |
| Language UCP ISO (Address) | CP ISO language of the user |
| Language UP ISO (Address) | P ISO language of the user |
| Last Name (Address) | Last name of the user |
| Location (Address) | Location name |
| Logon Language (Defaults) | Logon language for the user |
| Middle Name (Address) | Middle name of the user |
| Name Country (Address) | Name of the country |
| Nickname (Address) | Nickname of the user |
| Notes (Address) | Notes for the user |
| Other City (Address) | Name of the other city |
| Other City Number (Address) | Number of the other city |
| Pager/SMS List (Address) | Pager or SMS number list in the format pager_type#pager_number |
| Parameter List (Parameters) | Parameter list in the format prameter_ID=parameter_value |
| Pboxcity Number (Address) | Pbox number of the city |

| Attributes | Description |
|-------------------------------------|---------------------------------------|
| PCODE 1 Ext (Address) | Postal code 1 extension |
| PCODE 2 Ext (Address) | Postal code 2 extension |
| PCODE 3 Ext (Address) | Postal code 3 extension |
| PO Box (Address) | PO box number |
| PO Box City (Address) | PO box number of the city |
| PO Box City ISO (Address) | PO box number of the ISO city |
| PO Box Country (Address) | PO box number of the country |
| PO Box Region (Address) | PO box number of the region |
| PO Box Without Number (Address) | PO box without number |
| Postal Code (Address) | Postal code of the user |
| Postal Code 2 (Address) | 2nd postal code of the user |
| Postal Code 3 (Address) | 3rd postal code of the user |
| Prefix 1 (Address) | 1st prefix |
| Prefix 2 (Address) | 2nd prefix |
| Print Immediately (Defaults) | Print immediately flag for the user |
| Printer List (Address) | Print destination list |
| Region (Address) | Name of the region |
| Region Group (Address) | Group name of the region |
| Remote Communication List (Address) | Communication notes list |
| Remote Function Call List (Address) | Remote function call destination list |
| Remote Mail List (Address) | Remote mail list of the user |
| Room Number (Address) | Room number of the user |
| Room Number 2 (Address) | 2nd room number of the user |
| Search Term 2 P (Address) | 2nd search term P for the user |
| Search Term P (Address) | Search term P for the user |
| Search Term 1 (Address) | 1st search term for the user |
| Search Term 2 (Address) | 2nd search term for the user |
| Second Name (Address) | Second name of the user |
| Start Menu (Defaults) | Start menu for the user |
| Street Abbreviation (Address) | Street abbreviation for the user |
| Street Address (Address) | Street address of the user |
| Street Address 2 (Address) | Street address 2 of the user |
| Street Address 3 (Address) | Street address 3 of the user |

Schema attributes

| Attributes | Description |
|---------------------------------|---|
| Street Address 4 (Address) | Street address 4 of the user |
| Street Number (Address) | Street number of the user |
| Tax Jurisdiction Code (Address) | Tax jurisdiction code of the user |
| Telephone (Address) | Telephone number |
| Telephone Extension (Address) | Telephone extension number |
| Telephone List (Address) | Telephone number list |
| Teletex List (Address) | Teletex number list |
| Telex List (Address) | Telex number list |
| Time Format (Defaults) | Time format of the user |
| Time Zone (Address/Defaults) | Time zone of the user |
| Title (Address) | Title of the user |
| Title SPPL (Address) | Title SPPL of the user |
| Transportation Zone (Address) | Transportation zone of the user |
| URL (Homepage) List (Address) | URL (Homepage) address list in the format URI_type#URI_name |
| User Last Login | User last log in time. |
| Password in permanent mode | User password set in permanent mode. |
| Deactivate | User password deactivated. |
| User Name | User Name. |
| User Title (Address) | Title of the user |
| User Type (Logon Data) | Type of the user |
| User Valid From (Logon Data) | Valid from date for the user |
| User Valid To (Logon Data) | Valid to date for the user. |
| User Group (Logon Data) | User group of the user |
| X.400 List (Address) | Organization name list |
| Roles | Roles for user. |
| Profiles | Profiles for user. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|-------------|---------------------------|
| Name | Role/Profile name. |
| Type | Role/Profile type. |
| Description | Role/Profile description. |

| Attributes | Description |
|------------------|---|
| Child Roles | Sub Role/Profile list. |
| Long Description | Role/Profile long description. |
| Subsystem | System name for CUA system aggregation. |

Schema extension and custom attributes

The schema can be extended up to the extent of the fields within the structures provisioned by the SAP standard BAPI. The fields in the following structures will be provisioned:

- ADDRESS
- ALIAS
- COMPANY
- DEFAULTS
- LOGONDATA
- PASSWORD

Note: No custom attributes will be supported during provisioning.

Provisioning Policy attributes

This section lists the different policy attributes of SAP Connector.

Note: The attributes marked with * sign are the required attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

| Attributes | Description |
|-------------------------|---|
| User Name* | Name of the user to create. |
| Password | Password for the user. |
| Last Name* | Last name of the user. |
| Permanent Mode Password | Setting password in permanent mode. |
| Deactivate | Creating user in deactivated password mode. |

Additional information

This section describes the additional information related to the SAP Connector.

Entitlement validity period

A SAP role (activity group) can have a Start Date and an End Date. The ability to select or specify the same, while requesting an entitlement for an account, is absent in IdentityIQ currently. As a result, SAP assigns current system date and 31.12.9999 as Start Date and End Date by default to the user role respectively.

CUA support

By default the SAP Connector would not download data from CUA configured SAP System. In order to override this behaviour, the **CUASystem** configuration parameter must be checked in configuration parameter list.

Troubleshooting

1 - Distribution of a user to SAP CUA Subsystem

In a SAP CUA landscape, a SAP role or profile requires a SUBSYSTEM to distribute the user to. The facility to select or specify the same, while requesting an entitlement for an account, is absent in IdentityIQ.

Workaround: The subsystem name is prepended to the Account-Group while aggregating account-groups from a SAP CUA system. As a result, only a limited subset of subsystem and account-group combinations will be available while requesting entitlements, and thus distributing users, in a SAP CUA landscape.

Chapter 33: SailPoint IdentityIQ SAP Enterprise Portal Connector

The following topics are discussed in this chapter:

| | |
|--|-----|
| Overview | 265 |
| Supported features | 265 |
| Supported Managed Systems | 266 |
| Prerequisites | 266 |
| Administrator permission | 266 |
| Configuration parameters | 266 |
| Schema attributes | 267 |
| Account attributes | 267 |
| Group attributes | 268 |
| Provisioning Policy attributes | 268 |
| Create account attributes | 268 |
| Create Group attributes | 269 |

Overview

SAP Enterprise Portal integrates information and applications across the enterprise to provide an integrated single point of access to information, enterprise applications, and services both inside and outside an organization. The Portal also provides the tools to manage this knowledge, to analyse and interrelate it, and to share and collaborate. Enterprise Portal can provide users with a single, role-based entry point into legacy systems such as SAP R/3; as well as databases, documents, web content, other data sources and a variety of applications.

SailPoint IdentityIQ SAP Enterprise Portal Connector was developed to manage the following entities of SAP User Management Engine (UME):

- UME User
- UME Role
- Portal Role

Supported features

SailPoint IdentityIQ SAP Enterprise Portal Connector provides support for the following features:

- Account Aggregation
- Account Group Aggregation
- Change Password
- Create/Update/Delete/Enable/Disable Account
- Create/Update/Delete Account Group
- Add/Remove Entitlement
- Pass-through Authentication

Supported Managed Systems

Following versions of SAP Portal Servers are supported by the SAP Enterprise Portal Connector:

- SAP Portal 7.1
- SAP Portal 7.2
- SAP Portal 7.3

Prerequisites

The **sailpoint_ume.sda** file must be deployed on the SAP Enterprise Portal server which must be provisioned. Perform the following steps to deploy the **sailpoint_ume.sda** file:

1. Copy the SDA file from **(\$build)/integration/sap/dist** directory to a temporary directory on the SAP server.
2. Navigate to the home directory of SAP Enterprise Portal server
..\usr\sap\ep_instance_name\J02\j2ee\console on SAP server and execute **textconsole.bat**.
3. Run the following command:
>DEPLOY tmpDir\sailpoint_ume.sda (location of the file sailpoint_ume.sda)
where *tmpDir* is the temporary directory where the SDA file is extracted.

Administrator permission

The administrative account must have either one of the following permissions mentioned below for performing provisioning operation:

- pcd:portal_content/administrator/user_admin/user_admin_role
- pcd:portal_content/administrator/system_admin/system_admin_role
- pcd:portal_content/administrator/super_admin/super_admin_role
- SAP_J2EE_ADMIN

Configuration parameters

The following table lists the configuration parameters of SAP Enterprise Portal Connector:

| Parameters | Description |
|--------------------|--|
| UMWebService URL * | The url for the UMWebService. For example, http://HOST:PORT url can use either http or https. However, when using https you must also configure the portal server's and the application server's keystores. If you are upgrading your system from previous version edit the url which was used previously that is http://HOST:PORT/irj/servlet/prt/soap/UMWebService?style=rpc_enc , to the http://HOST:PORT url. |
| Username* | Name of the SAP Portal administrator. |
| Password* | Password of the administrator. |

| Parameters | Description |
|----------------|---|
| Account Filter | Enter the string representation of a sailpoint.object.Filter object. Any account object matching the filter will be filtered out of the dataset. Here's an example of a filterString that will filter out all objects where the uniqueId starts with USER.R3_DATASOURCE . When the uniqueId.startsWith('USER.R3_DATASOURCE') property is non-empty filtering happens on the IdentityIQ server side and does not filter on the SAP portal side. |
| Group Filter | Enter the string representation of a sailpoint.object.Filter object. Any roles object matching the filter will be filtered out of the dataset. Here's an example of a filterString that will filter out all objects from the displayName that starts with com.sap.pct . When the displayName.startsWith("com.sap.pct") property is non-empty filtering happens on the IdentityIQ server side and does not filter on the SAP portal side. |

Schema attributes

This section describes the different schema attributes.

Note: The attributes marked with * sign are the required attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|--|------------------------------|
| uniqueId | Users unique identification. |
| firstName | Users first name. |
| lastName | Users last name. |
| displayName | Users display name. |
| Company | Users company name. |
| Title | Users title. |
| uniqueName (Identity Name+ Display Name) | Users unique name. |
| City | Users city. |
| postalCode | Users postal address. |
| email | Users email address. |
| street | Users street. |
| state | Users state. |
| country | Users country. |
| zip | Users postal zip code. |
| fax | Users fax. |

Provisioning Policy attributes

| Attributes | Description |
|--------------|------------------------------|
| telephone | Users telephone number. |
| cellPhone | Users cell phone number. |
| department | Users department assigned. |
| salutation | Users salutation. |
| jobTitle | Users job title. |
| timeZone | Timezone of the user. |
| Language | Language of the user. |
| securityType | Users's security type. |
| lockStatus | User is locked or open. |
| roles | Role assigned to the user. |
| groups | Groups assigned to the user. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|--------------------------------|--------------------------------|
| displayName | Display name of the role. |
| uniqueName (Display Attribute) | Unique name of the role. |
| uniqueId (Identity Attribute) | Unique ID of the role. |
| Description | Description of the role. |
| userMembers | Users associated to the role. |
| groupMembers | Groups associated to the role. |

Provisioning Policy attributes

This section lists the different policy attributes of SAP Enterprise Portal Connector.

Note: The attributes marked with * sign are the required attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

| Account attribute | Description |
|-------------------|------------------------------|
| uniqueId | Users unique identification. |
| First Name | Users first name. |

| | |
|---------------|---------------------------|
| Last Name* | Users last name. |
| Display Name | Users display name. |
| company | Users company name. |
| Department | Users department. |
| Unique Name* | Users unique name. |
| Password* | Users password. |
| City | Users city. |
| Street | Users Street. |
| Email | Users email address. |
| City | Users city. |
| State | Users state. |
| Country | Users country. |
| Zip | Users postal zip code. |
| Fax | Users fax. |
| Tele Phone | Users telephone number. |
| Cell Phone | Users cell phone number. |
| Salutation | Users salutation. |
| JobTitle | Users job title. |
| Language | Language of the user. |
| Security Type | Users security type. |
| Lock Status | Users are locked or open. |

Create Group attributes

The following table lists the provisioning policy attributes for Update Account:

| Attributes | Description |
|---------------|--------------------------------|
| Role Name* | Display name of the role. |
| Description | Description of the role. |
| User Members | Users associated to the role. |
| Group Members | Groups associated to the role. |

Provisioning Policy attributes

Chapter 34: SailPoint IdentityIQ Tivoli Access Manager Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|-----|
| Overview | 271 |
| Supported features | 271 |
| Supported Managed System | 271 |
| Pre-requisites | 271 |
| Configuration parameters | 272 |
| Schema attributes | 273 |
| Account attributes | 273 |
| Group attributes | 273 |
| Provisioning Policy attributes | 274 |
| Create account attributes | 274 |
| Create group attributes | 274 |
| Additional information | 275 |
| Unstructured Target Collector | 275 |
| Troubleshooting | 275 |

Overview

SailPoint IdentityIQ Tivoli Access Manager Connector was developed to manage Users and their Entitlements through groups present in Tivoli Access Manager system from IdentityIQ.

Supported features

SailPoint IdentityIQ Tivoli Access Manager Connector provides support for the following features:

- Account Aggregation
- Account-Group Aggregation
- Passthrough Authentication
- Create/Delete Account
- Request/Remove Entitlement
- Create/Delete Group
- Change Password
- Removal of direct permissions (ACL membership)

Supported Managed System

SailPoint IdentityIQ Tivoli Access Manager Connector supports Tivoli Access Manager version 6.1.

Pre-requisites

1. Install the IBM Tivoli Access Manager Java Runtime component on the IdentityIQ server.

Configuration parameters

2. Install the following libraries in JRE's **lib/ext** directory:

- PD.jar
- Ibmjcefpis.jar
- Ibmjcefw.jar
- Ibmjceprovider.jar
- ibmjsseprovider2.jar
- ibmpkcs.jar
- local_policy.jar
- US_export_policy.jar

Note: Among the above files, PD.jar file can be found on Tivoli Access Manager java runtime installation and others are part of IBM's Java 1.5.

3. Tivoli Access Manager Authorization APIs uses the Java Authentication and Authorization Service (JAAS), for this the following changes are required in **java.security** file:
 - a. Specify the login file location: Point to the login configuration file from the **JAVA_HOME/jre/lib/security/java.security** file. For example, a sample entry from the **java.security** file might look like **login.config.url.1=file:\${java.home}/lib/security/login.pd**
 - b. Creating a login configuration file: Create **login.pd** on the mentioned location, if it does not exist add an entry as follows:

```
pd {
  com.tivoli.pd.jazn.PDLoginModule required;
};
```
 - c. Specify the policy file location: Point the policy file location from **JAVA_HOME/jre/lib/security/java.security** file. It displays as follows:**policy.url.1=file:\${java.home}/lib/security/java.policy**

Add a grant permission entry as follows:

```
permission javax.security.auth.AuthPermission "createLoginContext";"
```

4. Create configuration file needed by Tivoli Access Manager Connector with the help of **com.tivoli.pd.jcfg.SvrSslCfg** utility.

For example,

```
java com.tivoli.pd.jcfg.SvrSslCfg -action config -admin_id sec_master -admin_pwd
Sailpoint123 -appsvr_id server1 -host 172.16.21.157 -port 9080 -mode remote -policysvr
172.16.21.157:7135:1 -authzsvr 172.16.21.157:7136:2 -cfg_file C:\configfile -key_file
C:\keystore -cfg_action create
```

Configuration parameters

The following table lists the configuration parameters of Tivoli Access Manager Connector:

| Parameters | Description |
|-----------------|--|
| Admin Name* | Tivoli Access Manager administrator name. |
| Admin Password* | Password of the administrator. |
| Domain | The domain that is to be managed by the connector. |

| Parameters | Description |
|-------------------------|--|
| Configuration file URL* | A URL reference to configuration data file generated by <code>com.tivoli.pd.jcfg.SvrSslCfg</code> utility. |

Schema attributes

This section describes the different schema attributes.

Note: All the attributes marked with * sign are the mandatory attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|--------------------|--|
| userid | User ID of the user. |
| firstname* | The user's first name. |
| lastname* | The user's last name. |
| registryUID* | The account name stored in the user registry. |
| description | Text describing the user. |
| groups | The Access Manager groups that the user is a member of. |
| noPwdPolicy | Indicates whether a password policy is enforced. |
| ssoUser | Indicates whether the user has single sign-on abilities. |
| passwordvalid | The valid password. |
| accountValid | Indicates whether the account is disabled. |
| gsoWedCreds | gsoWedCreds. |
| gsoGroupCreds | Shows the list of gso group credentials assigned to a user. Will be shown as <userid>:<gso group name>. |
| importFromregistry | Indicates that the new user must be imported from registry server and not created in registry server. Note that the user must be present in the registry server. |

Group attributes

The following table lists the group attributes:

| Attributes | Description |
|--------------|---|
| GroupName* | Name of the group. |
| registryUID* | The group name stored in the user registry. |
| description | Text describing the Group. |

Provisioning Policy attributes

This section lists the different policy attributes of Tivoli Access Manager Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

| Attributes | Description |
|-----------------------|--|
| UserID | User ID for the user. |
| First Name | The user's first name. |
| Last Name | The user's last name. |
| registryUID | The account name stored in the user registry. |
| Description | Text describing user. |
| Password | Password for the user. |
| Password Valid | Indicates whether the password will be expired. |
| Account Valid | Indicates whether the account is disabled. |
| GSO User | Indicates whether the user has single sign-on abilities. |
| GSO Web Credentials | List of gso credentials to be given to the new user. It should be as follows: <gso name>:<userid>:<gso-password> |
| GSO Group Credentials | List of gso group credentials to be given to the new user. It should be as follows: <gso group name>:<userid >:<gso-password> |
| ImportFromRegistry | Indicates whether the user is to be imported from registry server. |
| No Password Policy | Indicates whether a password policy is enforced. |

Configuration settings

Field separator for **GSO Web Credentials** and **GSO Group Credentials** can be defined in application template using the debug page from IdentityIQ. Default delimiter is ':'.

For example, entry: <entry key="gso_field_seperator" value="#" />

The above example will set the field separator for mentioned attributes to '#'.

Note: The delimiter selected should not be a part of any of the subfields in the mentioned attribute. For above example character '#' should not be part of **gso name** or **userid** or **gso password** on **GSO Web Credentials** attribute.

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

| Attributes | Description |
|-------------|---|
| Name | Name of the group. |
| registryUID | The group name stored in the user registry. |
| Last Name | The user's last name. |
| Description | Text describing the user. |

Additional information

This section describes the additional information related to the Tivoli Access Manager Connector.

Unstructured Target Collector

Tivoli Access Manager uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with **account identityAttribute** for Accounts and **group identityAttribute** for AccountGroups. For more information on the **Unstructured Targets** tab, see "Unstructured Targets Tab" section of the *SailPoint IdentityIQ User's Guide*.

The Unstructured Targets functionality will be enabled for Tivoli Access Manager connector if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Tivoli Access Manager Target Collector supports aggregation of Access Control List (ACL). Access permissions on ACL will be correlated to IdentityIQ Users and Groups.

Following is the configuration parameter in **Unstructured Targets** tab for Tivoli Access Manager Connector:

| Attribute | Description |
|--|--|
| IBM Tivoli Access Manager Application Name | Name of the IBM Tivoli Access Manager application. |

Troubleshooting

1 - Connector not aggregating all accounts

When you have the LDAP user registry setup for Tivoli Access Manager, the Connector might not aggregate all accounts from Tivoli Access Manager.

Resolution: The Maximum search results is controlled by the following parameters:

- The **max-search-size** stanza entry in the [ldap] stanza of the **ldap.conf** configuration file:
To indicate that there is no limit, set the stanza entry **max_search_size** to 0.
For example: max-search-size = 0

Troubleshooting

Note: Restart the Tivoli Access Manager servers for the required changes.

- The **ibm-slapdSizeLimit** parameter in the Tivoli Directory Server server **slapd32.conf** or **ibmslapd.conf** configuration file:

To indicate there is no limit, set the size limit to 0.

For example: `ibm-slapdSizeLimit = 0`

Note: This parameter affects all LDAP searches.

Note: Ensure that both parameters are set to value greater than or equal to the total number of records in Tivoli Access Manager.

Chapter 35: SailPoint IdentityIQ Tenrox Connector

SailPoint IdentityIQ Tenrox Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------------|-----|
| Overview | 277 |
| Supported features | 277 |
| Supported Managed Systems | 277 |
| Pre-requisites | 278 |
| Administrator permission | 278 |
| Configuration parameters | 278 |
| Schema attributes | 278 |
| Account attributes | 278 |
| Provisioning Policy attributes | 279 |
| Troubleshooting | 280 |

Overview

Tenrox connector can be used to manage Tenrox user accounts and their permissions (security roles) from IdentityIQ.

Supported features

The Tenrox connector provides support for the following operations:

- Account Aggregation

Note: By default only the Tenrox users will be aggregated. If Tenrox template is also required along with the Tenrox Users, set the **IsGenericResource** application attribute as **true**.

- Create/Update/Delete Account
- Enable/Disable/Unlock Account
- Account Refresh
- Request Entitlement (Security Roles that are assigned to user are only fetched in and are available in Entitlement catalogue. Hence roles which are not assigned to any user would not be available in the catalogue.)
- Remove Entitlement (Security Role)

Note: Removing Entitlement will assign the site defined default entitlement (Security Role) to Tenrox User. (As Tenrox user can have only one Security role attached).

- Pass through authentication

Supported Managed Systems

SailPoint IdentityIQ Tenrox Connector supports Tenrox Software 2011 Release 3 Service Pack 5 (1051).

Pre-requisites

Tenrox application should be running and accessible.

Administrator permission

The Tenrox Administrator configured must have appropriate security role (that is, Administrator) for accessing and managing Tenrox user accounts. The administrator should have create/delete/modify permission on Tenrox system.

Configuration parameters

The Tenrox SOAP based web services is used to communicate with Tenrox System.

| Parameters | Description |
|-------------------|--|
| Tenrox URL | URL for accessing Tenrox (For example the format of the URL is follows: https://orgname.tenrox.net |
| Tenrox Admin User | User Name used for logging into Tenrox with sufficient rights (administrator security role). |
| Password | User's Password. |
| Organization Name | Organization name (case sensitive) which is subscribed. |

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

| Attributes | Description |
|-----------------|--|
| UniqueId | User's unique id |
| AddressLine1 | Address description |
| AddressLine2 | Address description |
| ApprovalGroup | Approve group which the user is a member of |
| FunctionalGroup | Functional group which the user is a member of |
| ResourceGroup | Resource group which the user is a member of |
| BadgeNumber | Punch card / bio metric card number |
| City | City name |

| Attributes | Description |
|-----------------------|---|
| DateHired | Date hired |
| DateOfBirth | Date of birth |
| Description | Description |
| Email | Email |
| FullName | Full name |
| IMSignin | Instant messenger sign in ID |
| Id | Employee Id |
| Language | Language |
| LastName | Last name |
| LoginName | Name which is used for logging into Tenrox |
| MaritalStatus | Marital status |
| MobileNumber | Mobile number |
| PostalCode | Postal code |
| Security | Role based on which access to Tenrox modules is defined |
| ServiceDate | Service date |
| SocialInsuranceNumber | Social Insurance number |
| State | State |
| TelephoneNumber | Telephone number |
| Title | Designation |
| IsDecommissioned | Permanent deactivation |
| IsSuspended | Temporary deactivation |

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Accounts:

| Parameters | Description |
|------------|---|
| Email | Email |
| First name | First name |
| Last name | Last name |
| Logon name | Name which is used for logging into Tenrox. |

Troubleshooting

1 - When creating a Tenrox user using password field from IdentityIQ, the user password cannot be set in Tenrox

When creating a Tenrox user using password field from IdentityIQ, the user password cannot be set in Tenrox as Tenrox requires hashed password value.

Resolution: Tenrox users created from IdentityIQ, have to use the forgot password link of Tenrox Application, to set their password or ask Tenrox administrator to set the password.

Section 2: Read Only Direct Connectors

This section contains the information on the following:

- SailPoint IdentityIQ Yammer Connector on page 283
- SailPoint IdentityIQ ALES Connector on page 287
- SailPoint IdentityIQ Logical Connector on page 289
- SailPoint IdentityIQ Delimited Connector on page 293
- SailPoint IdentityIQ LDIF Connector on page 297
- SailPoint IdentityIQ IBM Tivoli Identity Manager Connector on page 303
- SailPoint IdentityIQ SAP HR/HCM Connector on page 309
- SailPoint IdentityIQ SAP Portal-User Management Web Service Connector on page 311
- SailPoint IdentityIQ Sun IDM Connector on page 315
- SailPoint IdentityIQ Top Secret Connector on page 317
- SailPoint IdentityIQ UNIX Connector on page 331
- SailPoint IdentityIQ Mainframe Connector on page 333
- SailPoint IdentityIQ Novell Identity Manager Connector on page 335
- SailPoint IdentityIQ RACF Connector on page 341
- SailPoint IdentityIQ Rule Based Logical Connector on page 349
- SailPoint IdentityIQ SAP Connector on page 345

Chapter 36: SailPoint IdentityIQ Yammer Connector

The following topics are discussed in this chapter:

| | |
|-----------------------------------|-----|
| Overview | 283 |
| Pre-requisites | 283 |
| Configuration parameter | 283 |
| Schema attributes | 283 |
| Account attributes | 283 |
| Group attributes | 284 |

Overview

The Yammer Connector is a read only connector that retrieves accounts and groups from one or more networks on Yammer (Enterprise Social Network).

Pre-requisites

The user will be walked through the OAuth2 flow to generate the access token using the Cloud Commander and then pass it down to the IdentityIQ Yammer connector. The connector will use this Access Token to make calls to any Yammer REST API.

Configuration parameter

Access Token: A valid Access Token for the user is required which enables your application to access the user's information and take actions on their behalf. The application and user are verified with each API call by passing an access token along with each request.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

- **Account:** objects used when building identities Link objects.
- **Group:** schema used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 33—Account attributes](#)):

Table 33—Account attributes

| Attributes | Description |
|---------------|---|
| Admin | The user is an administrator in a specified network or not. |
| Department | The department of the user in the company. |
| Email | Email of the user . |
| EmailType | Type of Email (primary or secondary). |
| FullName | Full name of the user. |
| Groups | Groups to which user is a member of. |
| JobTitle | The job title of the user. |
| NetworkDomain | The Domain of the network of which the user is a member of. |
| NetworkID | The ID of the network of which user is a member of. |
| NetworkName | The name of the network of which user is a member of. |
| UserID | The ID of the user. |
| UserName | The username internally stored by Yammer for each user. |
| UserType | The retrieved identity is the user. |
| UserURL | The url which stores the property of user. |
| UserWebURL | The url for the web page of the user on Yammer. |
| Location | The location of the user. |
| Summary | The summary of the user. |

Group attributes

The following table lists the group attributes ([Table 34—Group attributes](#)):

Table 34—Group attributes

| Attributes | Description |
|------------------|--|
| GroupState | The group is active or not. |
| GroupType | The retrieved identity is group. |
| GroupWebURL | The URL of the web page fot that group on Yammer. |
| GroupPrivacy | The group is private or public. |
| GroupURL | The URL strores the property of the group. |
| GroupDescription | The description given for the creation of the group. |
| GroupFullName | The full name of the group. |
| GroupName | The name of the group. |
| GroupMembers | Contains all the members of the group. |

Table 34—Group attributes

| Attributes | Description |
|------------|----------------------|
| GroupID | The ID of the group. |

Schema attributes

Chapter 37: SailPoint IdentityIQ ALES Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 287 |
| Configuration parameters | 287 |
| Schema attributes | 288 |

Overview

This connector is designed to communicate with BEA's Aqualogic Enterprise Security Server. The integration uses the remote ALES Entitlement Query API.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The ALES connector uses the following connection attributes:

Table 35—ALES Connector - Configuration parameters

| Parameters | Description |
|--|--|
| filter | The accountfilter and groupfilter can be used to scope the iteration for accounts and groups. |
| javax.jdo.PersistenceManagerFactoryClass | The name of the concrete implementation of the javax.jdo.PersistenceManagerFactory that javax.jdo.JDOHelper.getPersistenceManagerFactory should create. For Kodo JDO, this should be kodo.jdbc.runtime.JDBCPersistenceManagerFactory or a custom extension of this type. |
| javax.jdo.option.ConnectionURL | The JDBC URL for the database. |
| javax.jdo.option.ConnectionDriverName | The full class name of either the JDBC java.sql.Driver, or a javax.sql.DataSource implementation to use to connect to the database. |
| javax.jdo.option.ConnectionUserName | The user name to use when connecting to the database. |
| javax.jdo.option.ConnectionPassword | The password for the user specified in the ConnectionUserName property. |
| kodo.jdbc.DBDictionary | A plugin string describing the kodo.jdbc.sql.DBDictionary to use for database interaction. Kodo typically auto-configures the dictionary based on the JDBC URL, but you may have to set this property explicitly if you are using an unrecognized driver, or to plug in your own dictionary for a database Kodo JPA/JDO does not support out-of-the-box. |
| kodo.Sequence | A plugin string describing the kodo.kernel.Seq implementation to use for the system sequence. |

Schema attributes

Table 35—ALES Connector - Configuration parameters

| Parameters | Description |
|------------|---|
| kodo.Log | A plugin string describing a com.solarmetric.log.LogFactory to use for logging. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 36—ALES Connector - Account Attributes

| Name | Description |
|---------------|--------------------|
| qualifiedName | Qualified UserName |
| groups | List of groups |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 37—ALES Connector - Group Attributes

| Name | Description |
|---------------|----------------------|
| qualifiedName | Qualified Group name |
| directUsers | List of users |

Chapter 38: SailPoint IdentityIQ Logical Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------------|-----|
| Overview | 289 |
| Configuration parameters | 289 |
| Schema attributes | 289 |
| Additional information | 290 |
| Logical Connector - Tiers Tab | 290 |
| Defining Logical Connectors | 292 |
| Logical Application Filtering | 292 |

Overview

The Logical connector was developed to create objects that function like applications in the IdentityIQ product, but that are actually formed based on the detection of accounts from other, or tier, applications in existing identity cubes.

For example, you might have one logical application that represents three other accounts on tier applications, an Oracle database, an LDAP authorization application, and a custom application for internal authentication. The logical application scans identities and creates an account on the logical application each time it detects the three required accounts on a single identity.

You can then use the single, representative account instead of the three separate accounts from which it is comprised for certification, reporting, and monitoring.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Logical applications do not have connection attributes, by default. If you have defined custom logical connectors there might be connection attributes on this tab.

Use this tab to test your logical application connection.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Logical applications enable you to pull schema attribute information from the tier applications from which it is compiled. When you use this feature the schema attribute information is automatically added to the attributes table and you can edit it as needed.

Click **New Tier Attribute** to display the Select Source Attribute dialog and select the tier application and attribute to pull into the logical application.

Additional information

This section describes the additional information related to the Logical Connector.

Logical Connector - Tiers Tab

This section contains the information that IdentityIQ uses to build the relationships between the tier applications that make up a logical application. For an identity to have an account on a logical application they must have the required, matching accounts on all tier applications. For example an identity, Lori Ferguson, might be represented by the attribute **dbid** on one tier and **username** on another. You must correlate those attributes, either manually or with a correlation rule, to create accounts on the logical application.

Add Tiers to a Logical Application

You must define the tier applications that are contained within the logical application and identify the application to be used as the primary tier application.

To add a tier application, select the application from the Select an Application drop-down list and click **Add Tier**. Click the arrow to right of the field to display all applications configured to work with IdentityIQ or type the first few letters of an application name to display a list of applications with names containing that letter string. You can add as many applications as required.

Specify the primary tier application by selecting it in the Primary Tier column. The primary tier application is the application containing all of the attributes to which the attributes on the other tiers will correlate. Every account on the logical application must have an account on the primary tier application. In some instances this might be a human resources application containing all of the identities in IdentityIQ. A logical application can only have one primary tier application.

To remove tier applications, select the application using the selection boxes in the left-most column and click **Remove Selected**.

Correlate Tier Application Attributes

Use the logical application tier attribute mapping, or correlation, panel to either manually map attributes for correlation or assign an existing correlation rule. For an identity to have an account on a logical application they must have the required, matching, accounts on all tier applications. Map the attributes on each application that should have matching values.

To manually map attributes on the tier applications do the following:

1. Select a non-primary tier application in the application list. The selected application is highlighted and any mapped correlation attributes are displayed in the attribute correlation panel.
If you select the primary tier application a note is displayed stating that no correlation is required on the primary tier.
2. Click **Add Attribute** to display a row in which to add the new attribute.
3. Click on the row to activate either the **Tier Attribute** or **Primary Tier Attribute** field.
4. Select an attribute from the drop-down list in both columns.
5. Click **Save Changes** or continue mapping attributes for the applications.

To use an existing correlation rule, open the Use Correlation Rule panel and select a rule from the **Correlation Rule** drop-down list. The rule should contain all of the attribute mapping required for this logical application.

The Tiers tab contains the following information:

Table 38—Logical Connector - Tier Applications

| Attribute | Description |
|------------------------|---|
| Account Rule | <p>Select an existing account rule from the drop-down list.</p> <p>The logical application rule defines the requirements that must be met before an identity is assigned an account on this logical application. Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.</p> |
| Provisioning Rule | <p>Select an existing provisioning rule from the drop-down list.</p> <p>The logical provisioning rule defines how provision requests for the logical application account or any of the accounts with which it is comprised are handled. Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.</p> |
| Application | The tier applications that make up the logical application. |
| Primary Tier | <p>Designate one tier application as the primary tier application. The primary tier application is the application containing all of the attributes to which the attributes on the other tiers will correlate. Every account on the logical application must have an account on the primary tier application. In some instances this might be a human resources application containing all of the identities in IdentityIQ.</p> <p>Note: A logical application can only have one primary tier application.</p> |
| Tier Attribute | Attributes from the selected tier application whose values must match the values of the associated attributes from the primary tier application. |
| Primary Tier Attribute | Attributes on the primary tier application to which the attribute values from the tier applications must match. |
| Account Matching | <p>Use account matching to select attributes and permissions from existing application tiers as the parameters for your logical application. This panel contains the following:</p> <p>Application Items — Click Add Attribute to include application attributes in your account matching parameters. Click Add Permission to include application permissions in you account matching parameters.</p> <p>Operation — choose the AND / OR operator to include multiple attributes / permissions</p> <p>Type — indicates either Attribute or Permission</p> <p>Application — indicates the application from which the attribute or permission is being matched</p> <p>Name — select an attribute from the drop-down list or input the permission name into the field</p> <p>Value — input the value of the attribute or permission</p> <p>Group/Ungroup/Delete Selected — use the check box to select line items on which to perform the respective action</p> |

Defining Logical Connectors

Use the following procedure to define a logical connector.

1. Define all tier applications.
2. Perform the following tasks on each tiered application:
 - a. Run aggregation task.
 - b. Run entitlement correlation task.
 - c. Scan for missing entitlements or define new managed entitlements.
3. Define the logical application
 - a. Define application tiers
 - b. Discover schema attributes from selected tier applications for editing.
 - c. Scan for missing entitlements using the filters from the selected tiered applications for editing.
4. Run aggregation task on your newly defined logical application.

Logical Application Filtering

Logical applications use the **Find missing entitlement** scan on the Managed Entitlements tab as filtering action using the Account Matching criteria provided on the Tiers tab. This gives a more focused starting point instead of using all of the entitlement values from the selected application tiers.

For example, a new logical application uses the “memberOf” attribute in Active Directory. There are likely thousands of values that are assigned in an enterprise. With specific criteria defined in Account Matching, only the values you are interested in appear for easier editing.

Chapter 39: SailPoint IdentityIQ Delimited Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 293 |
| Configuration parameters | 293 |
| Schema attributes | 295 |

Overview

The Delimited File connector is a rule driven connector. This connector has rules that can be customized to handle the complexity of the data that is being extracted.

This connector can be configured to enable the automatic discovery of schema attributes. See Schema attributes on page 295.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Delimited File connector uses the following account and group connection attributes. The account and group attributes are divided onto two tabs for ease and clarity of use. The group attributes are optional and the setting default to settings from the Account tab if they are not specifically defined.

Table 39—Delimited File Connector - Account Tab Descriptions

| Parameters | Descriptions |
|--------------------------------------|--|
| File: | |
| Parsing Type | Enter which type of parsing technique should be used when parsing the contents of the data file. Note: The parsing type is only applicable for Account Attributes. |
| File Path | Enter the path and name of the data file that should be parsed. |
| File Encoding | Specify the encoding that was used when saving the data file. If this is left blank the application's server default encoding will be used when parsing the file. |
| Delimiter | Enter the character that should be used as a delimiter. If the delimiter is a unicode character use the <code>\\u</code> format. For example, <code>\\u0009</code> is used to specify for the tab character. |
| File has column header on first line | Only available for Delimited Parsing Type. Select this option if the data file has a header defined on the first line of the file. |

Table 39—Delimited File Connector - Account Tab Descriptions

| Parameters | Descriptions |
|---|--|
| Fail on column length mismatch | Only available for Delimited Parsing Type. Select this option if you want the connector to fail if all of the columns are not part of each line. Sometimes the last token is left out of the data. If this is the case in your file, select this option. |
| Regular Expression | Only available for Regular Expression Parsing Type. Enter the regular expression using regular expression groups that can be used to break the data into tokens. |
| Columns | Enter the names of the columns that will be used while parsing the file. If you are using the Regular Expression Parsing Type, field is required. If you are using the Delimited Parsing Type, you only have to configure this field if there is not a header defined or you want to rename of the columns that will be used in the buildMap rule. |
| Transport: | |
| Note: Transport attributes only apply to Accounts. | |
| File Transport | Specify how the file will be transferred. If the file resides locally on the application server, select Local . |
| Host | Specify the host name where the file is located |
| User | Specify the username that will be used during the file transfer. |
| Password | Specify the password for the user that will be used during the file transfer. |
| Filtering: | |
| Number of lines to skip | Enter the number of lines to skip from the top of the data file before parsing begins. |
| Filter Empty | Select this option if you want to filter out any objects that parse but have no attributes. |
| Comment Character | Enter a comment character used in the data file. Any line starting with this character will be skipped. |
| Filter String | Enter the string representation of an filter object. Any object matching the filter will be filtered out of the dataset and will not be returned. For example, a filter that will filter out all objects from the Manufacturing department is written as follows: department == /&quot;Manufacturing/&quot;; |
| Merging: | |
| Data needs to be merged | Select this option if the data for a single object spans multiple lines. |
| Index Column | Enter the name of the index column that will be used when finding like objects in the dataset. |

Table 39—Delimited File Connector - Account Tab Descriptions

| Parameters | Descriptions |
|---|---|
| Data sorted by the indexColumn(s)? | Select this option if the data is sorted by the index columns. If the data is not sorted, an in-memory representation of the data is built and used. |
| Which Columns should be merged? | Enter the names of the columns from the file from which values should be merged. |
| Connector Rules: Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed. | |
| Build Map Rule | A rule that is called for each row in the data file. This rule is used to convert the string tokens from the data file into a java.util.Map object. If a rule is not specified the connector builds a map with the contents keyed by the column name. |
| PreIterate Rule | A rule that is called before the iteration process begins and provides a hook for things like checking the file, building an alternate feed, or returning a stream object. |
| PostIterate Rule | A rule that is called after the iteration process has completed. |
| Map To ResourceObject Rule | A rule that is called for each unique java.util.Map created from the data file. This rule's job is used to convert a java.util.Map object, built from the data file, into a ResourceObject . If a rule is not specified the connector builds a ResourceObject using the schema. |
| MergeMaps Rule | A rule that is called during merging for each row that has a matching index column. The rule will receive the existing map along with the newly parsed map that has to be merged. If a rule is not specified the connector builds a combined java.util.Map using the original object and merges the attributes specified in the mergeColumns configuration option. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

For delimited file connectors the schema is usually dictated by the data in the file. If this connector is configured to use the automatic discovery function and you've specified column names (**columnNames**, **group.columnNames**, **account.columnNames**), those names are used to populate the schema. If there is a header in the file and the **hasHeader** option is enabled the columns are pulled directly from the file and populate the schema. All automatically generated schema attributes are marked as type String.

Schema attributes

Chapter 40: SailPoint IdentityIQ LDIF Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 297 |
| Configuration parameters | 297 |
| Schema Attributes | 298 |

Overview

The LDIF connector is used to extract data from LDIF files. To help when the membership is not part of the account data there is an option that can be configured named “groupMembershipAttribute”. This configuration setting holds the name of the attribute from the group file which contains the list of its members. Add this attribute to account schema and mark it multi-valued.

The “groupMembershipAttribute” along with a group file must be configured for this feature to work. During account iteration the connector will read in the groups file to get the group->use mapping and adorn each account with their assigned groups as they are aggregated.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The LDIF connector uses the following configuration parameters:

| Parameters | Description |
|-------------------------|--|
| filetransport | local, ftp, scp |
| host | The host of the server to which you are connecting. |
| transportUser | The user to use with ftp and scp. Not valid with local. |
| transportUserPassword | The password to use with of ftp and scp. Not valid with local. |
| file | The fully qualified path to the file. |
| fileEncoding | Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: http://www.iana.org/assignments/character-sets If this field is empty, the default encoding (the value of file.encoding specified by the jvm) is used. |
| mapToResourceObjectRule | Rule that is called to override the transformation of the data from the Map<String,String> form into a ResourceObject . |
| filterString | Filter lines that match this string. |
| filterEmptyRecords | If activated, records that have no data are filtered. |

Schema Attributes

| Parameters | Description |
|--------------------------|---|
| preIterativeRule | <p>The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values.</p> <p>This rule is called after the file has been transferred, but before iteration over the objects in the file is started.</p> <p>For validation this rule can use the existing statistics stored by the postIterationRule during the last aggregation. The rule can compare the stored values with the new values to check for problems</p> |
| postIterativeRule | <p>The post-iterate rule can store away the configuration object and rename/delete the file if desired.</p> <p>This rule is called after aggregation has completed and ALL objects have been iterated.</p> |
| groupMembershipAttribute | Holds the name of the attribute from the group file which contains the list of its members. |

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 40—LDIF Connector - Account Attributes

| Name | Description |
|------------------|--|
| businessCategory | The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: "engineering", "finance", and "sales". |
| carLicense | This attribute type contains the license plate or vehicle registration number associated with the user. |
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute contains the distinguished name by which the user is known. |
| departmentNumber | This attribute contains a numerical designation for a department within your enterprise. |

Table 40—LDIF Connector - Account Attributes (Continued)

| Name | Description |
|--------------------------|--|
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |
| destinationIndicator | This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom. Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed. |
| displayName | This attribute contains the preferred name to be used for this person throughout the application. |
| employeeNumber | This attribute contains the numerical identification key for this person within your enterprise. |
| employeeType | This attribute contains a descriptive type for this user, for example, contractor, full time, or part time. |
| facsimileTelephoneNumber | This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute. |
| givenName | This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. Examples: "John", "Sue", and "David". |
| groups | This attribute type contains a list of groups of which this person is a member. Example: "Sales" or "Engineering" |
| homePhone | This attribute contains the employee's home phone number. |
| homePostalAddress | This attribute contains the employee's mailing address. |
| initials | This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute. Examples: "J. A." and "J". |
| internationalISDNNumber | This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. Example: "0198 444 444". |

Table 40—LDIF Connector - Account Attributes (Continued)

| Name | Description |
|----------------------------|---|
| l | This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. Examples: "Austin", "Chicago", and "Brisbane". |
| mail | This attribute type contains the RFC822 mailbox for the user. |
| manager | This attribute type contains the distinguished name of the manager to whom this person reports. |
| mobile | This attribute type contains the mobile telephone number of this person. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated". |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| pager | This attribute type contains the telephone number of this persons pager. |
| physicalDeliveryOfficeName | This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E". |
| postOfficeBox | This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27". |
| postalAddress | This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA". |
| postalCode | This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA. |
| preferredDeliveryMethod | This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone". |
| preferredLanguage | This attribute type contains the preferred written or spoken language of this person. |
| registeredAddress | This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA". |

Table 40—LDIF Connector - Account Attributes (Continued)

| Name | Description |
|---------------------------|---|
| roomNumber | This attribute type contains the room or office number or this persons normal work location. |
| secretary | This attribute type contains the distinguished name of this persons secretary. |
| seeAlso | This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object “cn=Elvis Presley,ou=employee,o=xyz\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.”. Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values. |
| sn | This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: “Smith”. |
| st | This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”. |
| street | This attribute type contains site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”. |
| telephoneNumber | This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. |
| teletexTerminalIdentifier | The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute. |
| telexNumber | This attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute |
| title | This attribute type contains the persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”. |
| uid | This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”. |
| objectClass | The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 41—LDIF Connector - Group Attributes

| Name | Description |
|--------------|--|
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| uniqueMember | This attribute type contains the groups to which this person is a unique member. |
| dn | This attribute type contains the directory path to the object. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated". |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| owner | This attribute type contains the distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is "cn=All Employees, ou=Mailing List,o=xyz, Inc.", is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): "cn=Human Resources Director,ou=employee,o=xyz, Inc.". |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Chapter 41: SailPoint IdentityIQ IBM Tivoli Identity Manager Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 303 |
| Configuration parameters..... | 303 |
| Schema attributes | 304 |

Overview

The IBM Tivoli Identity Manager connector uses the **groupMemberSearchDN** attribute as the starting point in the directory to start searching for ALL group memberships. The IBM Tivoli Identity Manager does not store a user's group references on the user so this connector must always do a separate query to return a list of all of the user's groups.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

This connector uses the following connection attributes:

Table 42—IBM Tivoli Identity Manager Connector - Configuration parameters

| Parameters | Description |
|---------------------|--|
| useSSL | Specifies if the connection is over ssl. |
| authorizationType | Translates to the Context.SECURITY_AUTHENTICATION property in the api. The attribute value is one of the following strings: none , simple , strong . |
| user | The user to connect as. Typically a DN string such as Administrator. |
| password | The password for the administrator account. |
| port | The port the server is listening through. |
| host | The host of the LDAP server. |
| searchScope | The depth to search the LDAP tree. OBJECT_SCOPE , ONELEVEL_SCOPE , and SUBTREE_SCOPE . |
| searchDN | The search starting point. This is a DN string. |
| iterateSearchFilter | An optional filter that can be added to the configuration to scope the objects returned when the iterateObjects method is called. |
| pageSize | The number of objects to get, per page, when iterating over large numbers of objects. The default is 500. |
| groupMemberSearchDN | Where to start in the tree when resolving a user's group membership. This is a DN String. |

Schema attributes

Table 42—IBM Tivoli Identity Manager Connector - Configuration parameters

| Parameters | Description |
|----------------------|--|
| filterString | This setting can be used to filter object as they are returned for an underlying application. Derived attributes can also be included in the filter. |
| groupMemberAttribute | The name of the attribute used to store group members. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 43—IBM Tivoli Identity Manager Connector - Account Attributes

| Name | Description |
|------------------|--|
| businessCategory | The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: "engineering", "finance", and "sales". |
| carLicense | This attribute type contains the license plate or vehicle registration number associated with the user. |
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute contains the distinguished name by which the user is known. |
| departmentNumber | This attribute contains a numerical designation for a department within your enterprise. |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Table 43—IBM Tivoli Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|--------------------------|--|
| destinationIndicator | <p>This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom.</p> <p>Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.</p> |
| displayName | This attribute contains the preferred name to be used for this person throughout the application. |
| employeeNumber | This attribute contains the numerical identification key for this person within your enterprise. |
| employeeType | This attribute contains a descriptive type for this user, for example, contractor, full time, or part time. |
| facsimileTelephoneNumber | This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute. |
| givenName | <p>This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute.</p> <p>Examples: "John", "Sue", and "David".</p> |
| groups | <p>This attribute type contains a list of groups of which this person is a member.</p> <p>Example: "Sales" or "Engineering"</p> |
| homePhone | This attribute contains the employee's home phone number. |
| homePostalAddress | This attribute contains the employee's mailing address. |
| initials | <p>This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute.</p> <p>Examples: "J. A." and "J".</p> |
| internationalISDNNumber | <p>This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute.</p> <p>Example: "0198 444 444".</p> |
| l | <p>This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute.</p> <p>Examples: "Austin", "Chicago", and "Brisbane".</p> |
| mail | This attribute type contains the RFC822 mailbox for the user. |

Table 43—IBM Tivoli Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|----------------------------|---|
| manager | This attribute type contains the distinguished name of the manager to whom this person reports. |
| mobile | This attribute type contains the mobile telephone number of this person. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| pager | This attribute type contains the telephone number of this persons pager. |
| physicalDeliveryOfficeName | This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E". |
| postOfficeBox | This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27". |
| postalAddress | This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA". |
| postalCode | This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA. |
| preferredDeliveryMethod | This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone". |
| preferredLanguage | This attribute type contains the preferred written or spoken language of this person. |
| registeredAddress | This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA". |
| roomNumber | This attribute type contains the room or office number or this persons normal work location. |
| secretary | This attribute type contains the distinguished name of this persons secretary. |

Table 43—IBM Tivoli Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|---------------------------|---|
| seeAlso | This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object “cn=Elvis Presley,ou=employee,o=xyz\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.”. Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values. |
| sn | This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: “Smith”. |
| st | This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”. |
| street | This attribute type contains site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”. |
| telephoneNumber | This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. |
| teletexTerminalIdentifier | The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute. |
| telexNumber | This attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute |
| title | This attribute type contains the persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”. |
| uid | This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”. |
| objectClass | The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Schema attributes

| Name | Description |
|-------------|---|
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| member | This attribute type contains the groups to which this person is a unique member. |
| dn | This attribute type contains the directory path to the object. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| owner | This attribute type contains the distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is "cn=All Employees, ou=Mailing List, o=xyz, Inc.", is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): "cn=Human Resources Director, ou=employee, o=xyz, Inc." |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Chapter 42: SailPoint IdentityIQ SAP HR/HCM Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 309 |
| Configuration parameters | 309 |
| Schema Attributes | 309 |

Overview

The SAP HR/HCM connector was developed to return all of the user information from the SAP HR/HCM system.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SAP HR/HCM connector uses the following connection attributes:

Table 44—SAP HM/HCM Connector - Configuration parameters

| Parameters | Description |
|-------------------------|--|
| host | Host on which the SAP Java Connector is running |
| user | SAP Administrator |
| password | SAP Administrator password |
| clientNumber | 001 in our install |
| systemID | SAP system ID |
| clientLanguage | Language used by the client SAP client |
| router | Router number |
| groupHeirarchyAttribute | Name of the group attribute that provides hierarchical information. Default value is Child Roles. |

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. Account objects are used when building identities Link objects.

Account attributes

Table 45—SAP HR/HCM Connector - Account Attributes

| Name | Description |
|-----------------------|--|
| Academic Grade | Academic grade attained by the person |
| Address | Address |
| Address Type | Address type; home, work |
| Address Type Code | Address type code |
| Admin Group | Administrative group to which the person belongs |
| Aristocratic Title | Aristocratic title that apply to this person |
| Birth Date | Date of birth |
| Birth Name | Name given at time of birth |
| Birth Place | Name or location of birth place |
| Business Area | Business area |
| Changed By | HR record last changed by |
| City | City in which the person is located |
| Co Area | Corporate area |
| Comp Code | Compensation code |
| Company Name | Name of the company by which they are employed |
| Contract | Contract: yes, no |
| Cost Center | Cost center with which they are associated |
| Country | Country in which they are located |
| Country Code | Country code |
| Country of Birth | Country in which they were born |
| Country of Birth Code | Country code for country in which they were born |
| District | District in which they are located or report |
| E Group | E-mail groups to which they belong |
| Email | E-mail address |
| Employee Number | Employee number |
| FirstName | Given name |
| Form of Address | Form of address; Miss, Mrs., Sir |
| FullName | Full, legal name |
| Fund | |
| Funds Center | |
| Gender | Gender |

Table 45—SAP HR/HCM Connector - Account Attributes (Continued)

| Name | Description |
|--------------------------|--|
| Gender Code | Gender code |
| Id Number | Identification number |
| Initials | Initials |
| Job | Title |
| Job Description | Description of their function |
| Known As | Nickname or preferred name |
| Language | Primary language |
| Language Code | Language code |
| Language ISO | Primary language ISO code |
| Last Changed On | Date on which this record was last changed or updated |
| LastName | Surname |
| LegPerson | |
| Marital Status Code | Code associated with the marital status of this person |
| Marital Status Since | Time period since the last change in marital status |
| MaritalStatus | Marital status |
| MiddleName | Middle name |
| Name | Full name |
| Name Format Indicator | |
| Nationality | Nationality |
| Nationality Code | Nationality code |
| Number of Children | Number of children |
| Org Key | Organizational key |
| Org Unit | Organizational unit |
| Organization Description | Organization description |
| P subArea | |
| Payarea | The area from which their pay is received |
| Payroll Admin | The payroll administrator associated with this person |
| Personal Admin | The personal administrator associated with this person |
| Personal Area | The personal area to which they report |
| Personal Number | Their personal number |
| Position | Title |
| Position Description | Description of job function |
| Reason Code | |

Table 45—SAP HR/HCM Connector - Account Attributes (Continued)

| Name | Description |
|-----------------------------|---|
| Religion | Religion |
| Religion Code | Religion code |
| Second Academic Grade | Secondary academic grade associated with this person |
| Second Address Line | Second line of address |
| Second Name Prefix | Secondary name prefix |
| Second Nationality | Secondary nationality |
| Second Nationality Code | Secondary nationality code |
| SecondName | Second name |
| State | State in which they are located |
| State Abbreviation | State abbreviation of the state in which they are located |
| State of Birth | State of birth |
| Sub E Group | |
| Supervisor | Supervisors name |
| Surname Prefix | Last name prefix |
| System user name (SY-UNAME) | User ID |
| Telephone | Telephone number and dialing code |
| Third Nationality | Third nationality |
| Time Admin | |
| Title | Function |
| Valid End | Valid end date for this record to terminate |
| Valid Start | Valid start date for this record to begin |
| Zip Code | Zip code for this person |

Chapter 43: SailPoint IdentityIQ Sun IDM Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 315 |
| Configuration parameters..... | 315 |

Overview

The Sun IDM connector was developed to return all of the user accounts and the capabilities defined in the Sun IDM system.

Note: The *opensaml.jar* file is required to integrate Sun IdM with IdentityIQ. This file is located in the WEB-INF/lib folder of your IdentityIQ installation directory.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Sun IDM connector uses the following connection attributes:

Table 46—Sun IDM Connector - Configuration parameters

| Parameters | Description |
|-------------------|--|
| user | The administrative account for IDM. |
| password | The password associated with the administrative account. |
| rpcRouterURL | The URL of the IDM environment. |
| IncludeOnlyAdmins | Return only administrators (Users with capabilities) when iterating over account objects. |
| filterTermRule | If you need more complex filters you can generate them in a rule that gets called by the option named filterTermRule . That rule needs to return a list of one or more FilterTerms. The rule gets the application and the schema and is called before the iteration begins. |

Configuration parameters

Chapter 44: SailPoint IdentityIQ Top Secret Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 317 |
| Configuration parameters | 317 |
| Schema Attributes | 318 |

Overview

The Top Secret connector was developed to read the TSSCFILE export.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Top Secret connector uses the following connection attributes:

Table 47—Top Secret Connector - Configuration parameters

| Parameters | Description |
|-------------------------|--|
| filetransport | local, ftp, scp |
| host | The host of the server to which you are connecting. |
| transportUser | The user to use with ftp and scp. Not valid with local. |
| transportUserPassword | The password to use with of ftp and scp. Not valid with local. |
| file | The fully qualified path to the file. |
| fileEncoding | Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: http://www.iana.org/assignments/character-sets If this field is empty, the default encoding (the value of file.encoding specified by the jvm) is used. |
| mapToResourceObjectRule | Rule that is called to override the transformation of the data from the Map<String,String> form into a ResourceObject . |
| filterString | Filter lines that match this string. |
| filterEmptyRecords | If activated, records that have no data are filtered. |

Table 47—Top Secret Connector - Configuration parameters

| Parameters | Description |
|---|---|
| preIterativeRule | <p>The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values.</p> <p>This rule is called after the file has been transferred, but before iteration over the objects in the file is started.</p> <p>For validation this rule can use the existing statistics stored by the postIterationRule during the last aggregation. The rule can compare the stored values with the new values to check for problems</p> |
| postIterativeRule | <p>The post-iterate rule can store away the configuration object and rename/delete the file if desired.</p> <p>This rule is called after aggregation has completed and ALL objects have been iterated.</p> |
| accountTypes | <p>The type of account use to connect to the server.</p> <p>The default value is USER, but additional values can be specified.</p> |
| groupTypes | <p>The group type of the connector.</p> <p>The default values are GROUP ACID and PROFILE ACID.</p> |
| Top Secret Attribute Customization Rule | <p>The rule used to extend the parsing capabilities to customer records or redefine existing record configurations.</p> <p>TopSecret records hold a record identifier and all of the fields that are part of that record. This rule use the TopSecretRecord and TopSecretField classes to work with that information.</p> |

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 48—Top Secret Connector - Account Attributes

| Attribute | Description |
|--------------------|---|
| XAUTH | The authorized level at which the user can access the resource. |
| VMMDISK | The VM minidisks owned by the user. |
| ACTION | Specifies which action(s) CA-Top Secret will take when access to a resource is attempted. |
| LOCK TIME(MINUTES) | The time interval before unattended or inactive terminals are locked. |
| LOCK TIME FACILITY | The lock time for all terminals connected to the specified facility. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|---------------------|---|
| LANGUAGE PREFERENCE | The language preference code the user. |
| VOLSER(OWNED) | The volumes to which the user has access. |
| NAME | Identifies the ACID name. Names can be up to 32 characters in length, must be surrounded by single quotes if embedded with blanks, and can use letters, numbers, and special characters. |
| SITRAN | Specifies which CICS transaction CA-Top Secret automatically executes after an ACID successfully signs on to a facility. Note: If a SITRAN is added to an ACID that already has a CICS transaction defined, the transaction is replaced. |
| HOME | Defines the initial directory pathname. This is the initial directory used when a user enters the OMVS command or enters the ISPF shell. The HOME keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If HOME isn't defined, OpenEdition MVS sets the initial directory for the user to the root directory. HOME is optional. |
| XA ACID | XAUTH Resource Class Name |
| MULTIPW | Used to assign or remove multiple password attributes, which means ACIDs need a different password to access each facility. |
| NOADSP | Used to prevent data sets, created by an ACID, from being automatically secured by MVS by setting the RACF bit. NOADSP is used to define an ACID that will be used to create data sets that cannot be automatically protected by CA-Top Secret. |
| AUDIT | Used to allow an audit of ACID activity. |
| NOPWCHG | To prevent ACIDs from changing passwords at either signon or initiation. |
| OIDCARD | Used to support the physical identification of users through operator identification cards. |
| TRACE | Used to activate a diagnostic trace on all ACID activity (initiations, resource access, violations, user's security mode, etc.) |
| SUSPEND | Used to prevent ACIDs from accessing the system when a violation occurs. |
| MRO | Used to support the use of the multi-region option. |
| CONSOLE | Used to grant or remove an ACID's ability to modify control options. For VM, options are modified via the TSS MODIFY command only. With VSE and OS/390, options are modified at the O/S console or via the TSS MODIFY command function. |
| GAP | Used to specify that a profile will become, or will cease to be, globally administrable. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|-----------------------|---|
| DUFXTR | Used to add or remove the DUFXTR attribute to an ACID. DUFXTR enables an ACID to use a RACROUTE REQUEST=AUTH (RACHECK) macro or the CA-Top Secret Application Interface to extract installation data (INSTDATA) or field data from a Security Record. DUFXTR is a component of the CA-Top Secret Dynamic Update Facility (DUF). |
| DUFUPD | Used to add or remove the DUFUPD attribute to an ACID. DUFUPD enables an ACID to use the CA-Top Secret Application Interface to update the installation data (INSTDATA) or field data from a Security Record. DUFUPD is a component of the CA-Top Secret Dynamic Update Facility (DUF). |
| TSOMPW | Used to support multiple TSO UADS passwords, on a user-by-user basis. |
| NOATS | Used to prevent an ACID in CICS and CA-IDMS from signing on via ATS (Automatic Terminal Signon). |
| ACEDEFAU | |
| ASUSPEND | Used to remove the suspension of an ACID that was suspended for administrative reasons. |
| XA DATASET | |
| WHO HAS RESOURCE | |
| PROFILE ACID | Used to assign profiles to an ACID. |
| PASSWORD | Used to assign a password, along with values that control its use, to a previously defined ACID. |
| PASSWORD EXPIRES DATE | The date, in string format, that the password expires. |
| PASSWORD INTERVAL | The interval in which the password must be changed. |
| PASSWORD FACILITY | The facility name applied to ACIDs with the multipw attribute. |
| OPIDENT | Used to assign or remove a CICS operator identification value that is equal to the ACID's OPIDENT entry in the CICS SNT (Signon Table). The OPIDENT value is placed into the ACID's TCT at signon. |
| OPPTY | Used to assign or remove a CICS operator priority from the associated ACID. The OPPTY value is placed into the ACID's TCT (Terminal Control Table) at signon. |
| PROGRAM | Used to secure system programs and utilities. |
| WHOHAS ADMIN | Used to determine who has administrative authority on the application. |
| ACIDS2 | |
| SOURCES | |
| TSOLPROC | Used to provide a default procedure to be used for TSO logon. The one- to eight-character logon procedure name. Procedure names are also TSO-related resources and the user must be permitted to any procedure name with which he attempts to log on. |
| DIV ACID | Specifies the Division ACID to which the ACID is attached. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|---------------------------|---|
| DIV NAME | The name assigned to the ACID within the zone. |
| SUSPENDED | The date, in string format, that the suspension ends. |
| WHOHAS XAUTH | A list of resources that may be accessed by the ACID shown in the command, the level at which the ACID may access the resource, and the owner of the resource. |
| TSOUNIT | The default unit name to be used for dynamic allocations under TSO. The one- to eight-character unit (device) name for dynamically allocated data sets. The name must be a defined generic unit class name at the installation. This field is not alterable by the user at logon and is not required for successful logon. |
| PHYSKEY | PHYSKEY (physical security key) supports external authentication devices. |
| ACID WITHIN DEPT/DIV/ZONE | Used to specify department, division, and zone to include. |
| DATE CREATED | The date on which the ACID was created. |
| DATE LAST MODIFIED | The date on which the ACID was last modified. |
| TIME LAST MODIFIED | The time at which the ACID was last modified. |
| ROOM NUMBER | The room number assigned to the ACID. |
| MISC2 | Used to give, or to remove, a CA-Top Secret administrator's authority to perform one or more administrative functions. |
| ACCESSLEVELS | |
| MISC8 | Used to give, or to remove, a CA-Top Secret administrator's authority to list the contents of the RDT, FDT or STC or to use the ASUSPEND administrative function. |
| XA MINIDISK | The minidisk authorization information for the ACID. |
| SCOPE | Used to give CA-Top Secret administrators the authority, or to remove their authority, to assign the SCOPE of an LSCA. |
| DIGITAL CERT NAME | The name of the digital certification. |
| DEPARTMENT | The Department ACID to which the ACID is attached. |
| DLFTGRP | The default group for the ACID. |
| WHO OWNS RESOURCE | The resources owned by the ACID. |
| TSOOPT | The default options that a TSO user may specify at logon. |
| WANAME | The person to whom SYSOUT information should be delivered for this ACID. |
| XA | |
| SYSID | The SYSID (which is actually the SMFID) that the authorizations for the ACID apply to. |
| BUILDING | The building in which the ACID is located. |
| TSOCOMMAND | Default commands issued upon login of the ACID. |
| DIGITAL CERT STARTS | The date, in string format, that the digital certification starts. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|----------------------|---|
| XAUTH LIBRARY | The libraries for which the ACID has authority. |
| WHOHAS FACILITY | Returns facility information for the ACID. |
| RESOURCE CLASS NAME | The resource class for which the ACID has authority. |
| FCT/PREFIX(OWNED) | |
| FACILITIES | The facilities to which the user has access. |
| TSOHCCLASS | The default hold class for TSO generated JCL for TSO the user. |
| DIGITAL CERT EXPIRES | The date, in string format, when the digital certification expires. |
| ZONE ACID | The Zone ACID to which the ACID is attached. |
| ZONE NAME | The name assigned to the ACID within the zone. |
| ADDRESS1 | Physical address for the ACID. |
| XAUTHDAYS | Days of the week the ACID is authorized on this application. |
| ACID TYPE | The ACID type, for example zone, division, or department. |
| ACID SIZE | The size of the ACID. |
| RESTRICT | |
| ADDRESS4 | Alternative physical address for the ACID. |
| NODSNCHK | To specify that no data set name check will be performed. That is, CA-Top Secret will bypass all data set access security checks. All data set access will be audited. |
| NOVOLCHECK | |
| NOLCFCHK | Used to allow an ACID to execute any command or transaction for all facilities, regardless of LCF (Limited Command Facility) restrictions. No auditing is done. |
| NOSUBCHK | Used to allow an ACID to bypass alternate ACID usage as well as all job submission security checking. Thus, associated ACIDs may submit all jobs regardless of the (derived) ACID on the job card being submitted. |
| NORESCHK | Used to allow an ACID to bypass security checking, including auditing, for all owned resources except data sets and volumes. |
| NOVMDCHK | Used to allow an ACID to bypass all checking for minidisk links. All links will be audited. NOVMDCHK is intended only to be applied to special products such as DASD space managers, which may link to many minidisks. |
| NOSUSPEND | Used to allow an ACID to bypass suspension due to violations. |
| TSODEST | The default destination identifier for TSO generated JCL for TSO users. |
| XA VOLUMN | |
| TSODEFPRFG | The default TSO performance group. |
| RESOURCE CLASS NAME2 | An additional resource class for which the ACID has authority. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|-------------------|---|
| MISC1 | A CA-Top Secret administrator's authority to perform one or more administrative functions. |
| GID | IDs of the groups to which the ACID belongs. |
| TSOUDATA | The site-defined data fields for a TSO user. |
| ACCESSLEVELS2 | |
| DSN/PREFIX(OWNED) | |
| TSOMSIZE | The maximum region size (in kilobytes) that the TSO user can specify at logon. |
| EXPIRES | The date on which the ACID expires. |
| TSOSCLASS | The default SYSOUT class for TSO generated JCL for the TSO users. |
| XAUTH FAC | |
| DEPT ACID | The Department ACID to which the ACID is attached. |
| DEPT NAME | The name assigned to the ACID within the department. |
| DATE LAST USED | Date the ACID was last used. |
| TIME LAST USED | Time the ACID was last used. |
| CPU | Name of the CPU on which the ACID was used. |
| FAC | System facilities defined to CA-Top Secret: BATCH, STC, TSO, IMS, CICS, NCCF, CA-Roscoe, WYLBUR, or any installation-defined facility. |
| COUNT | |
| SEGMENT | Used to allow TSS administrators to list data about fields in a specific segment. |
| RESOURCES | |
| TSOJCLASS | The default job class for TSO generated job cards from TSO users. |
| ADMIN BY | |
| XAUTH MODE | |
| TSOLACCT | The default account number used for TSO logon. |
| TSOLSIZE | The default region size (in kilobytes) for TSO. |
| LISTDATA | |
| OMVSPGM | The user's OpenEdition MVS shell program. This is the first program started when the OMVS command is entered, or when an OpenEdition MVS batch job is started using the BPXBATCH program. |
| SMSSTOR | The default storage keyword for the ACID. |
| UID | The unique user ID for the ACID. |
| ADDRESS3 | Alternative physical address for the ACID. |
| XAUTH PRIVPGM | The program pathing, if privileged program is in use. |
| TIME ZONE | The time zone attached to the ACID. |

Table 48—Top Secret Connector - Account Attributes (Continued)

| Attribute | Description |
|---------------------|---|
| MASTER FACILITY | |
| LCF FACILITY | |
| FACILITY NAME | |
| FACILITY UNTIL DATE | |
| INSTDATA | Used to record or remove information about an ACID. Up to 255 characters of information about an associated ACID may be used for convenient record keeping, or for interrogation by a user-written Installation Exit. |
| ADDRESS2 | Alternative physical address for the ACID. |
| GROUP ACID | The Group ACID to which the ACID is attached. |
| TSOMCLASS | The default message class for TSO generated JCL for TSO users. |
| MISC9 | To give, or to remove, a TSS administrator's authority to perform one or more high-level administrative functions. |

Table 49—Top Secret Connector - Group Attributes

| Attribute | Description |
|---------------------|---|
| XAUTH | The authorized level at which the user can access the resource. |
| VMMDISK | The VM minidisks owned by the user. |
| ACTION | Specifies which action(s) CA-Top Secret will take when access to a resource is attempted. |
| LOCK TIME(MINUTES) | The time interval before unattended or inactive terminals are locked. |
| LOCK TIME FACILITY | The lock time for all terminals connected to the specified facility. |
| LANGUAGE PREFERENCE | The language preference code the user. |
| VOLSER(OWNED) | The volumes to which the user has access. |
| NAME | Identifies the ACID name. Names can be up to 32 characters in length, must be surrounded by single quotes if embedded with blanks, and can use letters, numbers, and special characters. |
| SITRAN | Specifies which CICS transaction CA-Top Secret automatically executes after an ACID successfully signs on to a facility. Note: If a SITRAN is added to an ACID that already has a CICS transaction defined, the transaction is replaced. |
| HOME | Defines the initial directory pathname. This is the initial directory used when a user enters the OMVS command or enters the ISPF shell. The HOME keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If HOME isn't defined, OpenEdition MVS sets the initial directory for the user to the root directory. HOME is optional. |
| XA ACID | XAUTH Resource Class Name |

Table 49—Top Secret Connector - Group Attributes (Continued)

| Attribute | Description |
|------------------|---|
| MULTIPW | Used to assign or remove multiple password attributes, which means ACIDs need a different password to access each facility. |
| NOADSP | Used to prevent data sets, created by an ACID, from being automatically secured by MVS by setting the RACF bit. NOADSP is used to define an ACID that will be used to create data sets that cannot be automatically protected by CA-Top Secret. |
| AUDIT | Used to allow an audit of ACID activity. |
| NOPWCHG | To prevent ACIDs from changing passwords at either signon or initiation. |
| OIDCARD | Used to support the physical identification of users through operator identification cards. |
| TRACE | Used to activate a diagnostic trace on all ACID activity (initiations, resource access, violations, user's security mode, etc.) |
| SUSPEND | Used to prevent ACIDs from accessing the system when a violation occurs. |
| MRO | Used to support the use of the multi-region option. |
| CONSOLE | Used to grant or remove an ACID's ability to modify control options. For VM, options are modified via the TSS MODIFY command only. With VSE and OS/390, options are modified at the O/S console or via the TSS MODIFY command function. |
| GAP | Used to specify that a profile will become, or will cease to be, globally administrable. |
| DUFXTR | Used to add or remove the DUFXTR attribute to an ACID. DUFXTR enables an ACID to use a RACROUTE REQUEST=AUTH (RACHECK) macro or the CA-Top Secret Application Interface to extract installation data (INSTDATA) or field data from a Security Record. DUFXTR is a component of the CA-Top Secret Dynamic Update Facility (DUF). |
| DUFUPD | Used to add or remove the DUFUPD attribute to an ACID. DUFUPD enables an ACID to use the CA-Top Secret Application Interface to update the installation data (INSTDATA) or field data from a Security Record. DUFUPD is a component of the CA-Top Secret Dynamic Update Facility (DUF). |
| TSOMPW | Used to support multiple TSO UADS passwords, on a user-by-user basis. |
| NOATS | Used to prevent an ACID in CICS and CA-IDMS from signing on via ATS (Automatic Terminal Signon). |
| ACEDEFAU | |
| ASUSPEND | Used to remove the suspension of an ACID that was suspended for administrative reasons. |
| XA DATASET | |
| WHO HAS RESOURCE | |
| PROFILE ACID | Used to assign profiles to an ACID. |

Table 49—Top Secret Connector - Group Attributes (Continued)

| Attribute | Description |
|---------------------------|---|
| PASSWORD | Used to assign a password, along with values that control its use, to a previously defined ACID. |
| PASSWORD EXPIRES DATE | The date, in string format, that the password expires. |
| PASSWORD INTERVAL | The interval in which the password must be changed. |
| PASSWORD FACILITY | The facility name applied to ACIDs with the multipw attribute. |
| OPIDENT | Used to assign or remove a CICS operator identification value that is equal to the ACID's OPIDENT entry in the CICS SNT (Signon Table). The OPIDENT value is placed into the ACID's TCT at signon. |
| OPPRTY | Used to assign or remove a CICS operator priority from the associated ACID. The OPPRTY value is placed into the ACID's TCT (Terminal Control Table) at signon. |
| PROGRAM | Used to secure system programs and utilities. |
| WHOHAS ADMIN | Used to determine who has administrative authority on the application. |
| ACIDS2 | |
| SOURCES | |
| TSOLPROC | Used to provide a default procedure to be used for TSO logon. The one- to eight-character logon procedure name. Procedure names are also TSO-related resources and the user must be permitted to any procedure name with which he attempts to log on. |
| DIV ACID | Specifies the Division ACID to which the ACID is attached. |
| DIV NAME | The name assigned to the ACID within the zone. |
| SUSPENDED | The date, in string format, that the suspension ends. |
| WHOHAS XAUTH | A list of resources that may be accessed by the ACID shown in the command, the level at which the ACID may access the resource, and the owner of the resource. |
| TSOUNIT | The default unit name to be used for dynamic allocations under TSO. The one- to eight-character unit (device) name for dynamically allocated data sets. The name must be a defined generic unit class name at the installation. This field is not alterable by the user at logon and is not required for successful logon. |
| PHYSKEY | PHYSKEY (physical security key) supports external authentication devices. |
| ACID WITHIN DEPT/DIV/ZONE | Used to specify department, division, and zone to include. |
| DATE CREATED | The date on which the ACID was created. |
| DATE LAST MODIFIED | The date on which the ACID was last modified. |
| TIME LAST MODIFIED | The time at which the ACID was last modified. |
| ROOM NUMBER | The room number assigned to the ACID. |

Table 49—Top Secret Connector - Group Attributes (Continued)

| Attribute | Description |
|----------------------|---|
| MISC2 | Used to give, or to remove, a CA-Top Secret administrator's authority to perform one or more administrative functions. |
| ACCESSLEVELS | |
| MISC8 | Used to give, or to remove, a CA-Top Secret administrator's authority to list the contents of the RDT, FDT or STC or to use the ASUSPEND administrative function. |
| XA MINIDISK | The minidisk authorization information for the ACID. |
| SCOPE | Used to give CA-Top Secret administrators the authority, or to remove their authority, to assign the SCOPE of an LSCA. |
| DIGITAL CERT NAME | The name of the digital certification. |
| DEPARTMENT | The Department ACID to which the ACID is attached. |
| DLFTGRP | The default group for the ACID. |
| WHO OWNS RESOURCE | The resources owned by the ACID. |
| TSOOPT | The default options that a TSO user may specify at logon. |
| WANAME | The person to whom SYSOUT information should be delivered for this ACID. |
| XA | |
| SYSID | The SYSID (which is actually the SMFID) that the authorizations for the ACID apply to. |
| BUILDING | The building in which the ACID is located. |
| TSOCOMMAND | Default commands issued upon login of the ACID. |
| DIGITAL CERT STARTS | The date, in string format, that the digital certification starts. |
| XAUTH LIBRARY | The libraries for which the ACID has authority. |
| WHOHAS FACILITY | Returns facility information for the ACID. |
| RESOURCE CLASS NAME | The resource class for which the ACID has authority. |
| FCT/PREFIX(OWNED) | |
| FACILITIES | The facilities to which the user has access. |
| TSOHCLASS | The default hold class for TSO generated JCL for TSO the user. |
| DIGITAL CERT EXPIRES | The date, in string format, when the digital certification expires. |
| ZONE ACID | The Zone ACID to which the ACID is attached. |
| ZONE NAME | The name assigned to the ACID within the zone. |
| ADDRESS1 | Physical address for the ACID. |
| XAUTHDAYS | Days of the week the ACID is authorized on this application. |
| ACID TYPE | The ACID type, for example zone, division, or department. |
| ACID SIZE | The size of the ACID. |
| RESTRICT | |

Table 49—Top Secret Connector - Group Attributes (Continued)

| Attribute | Description |
|----------------------|---|
| ADDRESS4 | Alternative physical address for the ACID. |
| NODSNCHK | To specify that no data set name check will be performed. That is, CA-Top Secret will bypass all data set access security checks. All data set access will be audited. |
| NOVOLCHECK | |
| NOLCFCHK | Used to allow an ACID to execute any command or transaction for all facilities, regardless of LCF (Limited Command Facility) restrictions. No auditing is done. |
| NOSUBCHK | Used to allow an ACID to bypass alternate ACID usage as well as all job submission security checking. Thus, associated ACIDs may submit all jobs regardless of the (derived) ACID on the job card being submitted. |
| NORESCHK | Used to allow an ACID to bypass security checking, including auditing, for all owned resources except data sets and volumes. |
| NOVMDCHK | Used to allow an ACID to bypass all checking for minidisk links. All links will be audited. NOVMDCHK is intended only to be applied to special products such as DASD space managers, which may link to many minidisks. |
| NOSUSPEND | Used to allow an ACID to bypass suspension due to violations. |
| TSODEST | The default destination identifier for TSO generated JCL for TSO users. |
| XA VOLUMN | |
| TSODEFPRFG | The default TSO performance group. |
| RESOURCE CLASS NAME2 | An additional resource class for which the ACID has authority. |
| MISC1 | A CA-Top Secret administrator's authority to perform one or more administrative functions. |
| GID | IDs of the groups to which the ACID belongs. |
| TSOUDATA | The site-defined data fields for a TSO user. |
| ACCESSLEVELS2 | |
| DSN/PREFIX(OWNED) | |
| TSOMSIZE | The maximum region size (in kilobytes) that the TSO user can specify at logon. |
| EXPIRES | The date on which the ACID expires. |
| TSOSCLASS | The default SYSOUT class for TSO generated JCL for the TSO users. |
| XAUTH FAC | |
| DEPT ACID | The Department ACID to which the ACID is attached. |
| DEPT NAME | The name assigned to the ACID within the department. |
| DATE LAST USED | Date the ACID was last used. |
| TIME LAST USED | Time the ACID was last used. |
| CPU | Name of the CPU on which the ACID was used. |

Table 49—Top Secret Connector - Group Attributes (Continued)

| Attribute | Description |
|---------------------|---|
| FAC | System facilities defined to CA-Top Secret: BATCH, STC, TSO, IMS, CICS, NCCF, CA-Roscoe, WYLBUR, or any installation-defined facility. |
| COUNT | |
| SEGMENT | Used to allow TSS administrators to list data about fields in a specific segment. |
| RESOURCES | |
| TSOJCLASS | The default job class for TSO generated job cards from TSO users. |
| ADMIN BY | |
| XAUTH MODE | |
| TSOLACCT | The default account number used for TSO logon. |
| TSOLSIZE | The default region size (in kilobytes) for TSO. |
| LISTDATA | |
| OMVSPGM | The user's OpenEdition MVS shell program. This is the first program started when the OMVS command is entered, or when an OpenEdition MVS batch job is started using the BPXBATCH program. |
| SMSSTOR | The default storage keyword for the ACID. |
| UID | The unique user ID for the ACID. |
| ADDRESS3 | Alternative physical address for the ACID. |
| XAUTH PRIVPGM | The program pathing, if privileged program is in use. |
| TIME ZONE | The time zone attached to the ACID. |
| MASTER FACILITY | |
| LCF FACILITY | |
| FACILITY NAME | |
| FACILITY UNTIL DATE | |
| INSTDATA | Used to record or remove information about an ACID. Up to 255 characters of information about an associated ACID may be used for convenient record keeping, or for interrogation by a user-written Installation Exit. |
| ADDRESS2 | Alternative physical address for the ACID. |
| GROUP ACID | The Group ACID to which the ACID is attached. |
| TSOMCLASS | The default message class for TSO generated JCL for TSO users. |
| MISC9 | To give, or to remove, a TSS administrator's authority to perform one or more high-level administrative functions. |

Schema Attributes

Chapter 45: SailPoint IdentityIQ

UNIX Connector

The following topics are discussed in this chapter:

| | |
|--------------------------------|-----|
| Overview | 331 |
| Configuration parameters | 331 |
| Schema attributes | 331 |

Overview

The UNIX connector was developed to read and parse the **passwd** and **group** file from UNIX servers to build identities and groups. Since this connector is file based, there is some synergy between the UNIX and Delimited File connector.

Depending on your application configuration, IdentityIQ determines login success by authenticating using the ftp or scp service with the provided login credentials. Therefore, the **passwdfile** attribute of the UNIX application must be the same password file used by the system for authentication. This password file is typically **/etc/passwd**, but might be different in an environment where NIS is used.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The UNIX Database connector uses the following connection attributes:

Table 50—UNIX Connector - Configuration parameters

| Parameters | Description |
|-----------------------|---|
| host | The host of the server to which you are connecting. |
| filetransport | local, ftp, scp |
| transportUser | The user to use with ftp and scp. Not valid with local. |
| transportUserPassword | The password to use with of ftp and scp. Not valid with local. |
| passwdfile | The fully qualified path to the passwd file. |
| groupfile | The fully qualified path to the group file. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 51—UNIX Connector - Account Attributes

| Attribute | Description |
|-----------|---|
| homedir | The path to the user's home directory on the host system. The home directory is the directory in which the user keeps personal files such as initialization files and mail. |
| shell | The shell, or program, preferred by the user for accessing the command line interface. |
| info | The information pertaining to the user. |
| groups | The groups to which the user belongs. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 52—UNIX Connector - Group Attributes

| Attribute | Description |
|-----------|--|
| groupname | A name associated with the group. The group names are listed in the first comma-delimited field of the groups text file. |
| groupid | A group id used to identify the group. The group ids are listed in the third comma-delimited field of the groups text file. |
| members | A comma-delimited list of users who are members of the group. Members are listed in the forth comma-delimited field of the groups text file. |

Chapter 46: SailPoint IdentityIQ Mainframe Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 333 |
| Configuration parameters..... | 333 |
| Schema attributes | 334 |

Overview

This connector uses a technique called screen scraping and each deployment must write Rules to drive the login/logout/fetch accounts. The connector parses the screens and emulates the user during the interaction. On some legacy systems screen scraping is the only way to get to the data needed by IdentityIQ. Each Mainframe connector requires a lot of hands on configuration, because the Rules that drive this connector are very specific to the application on which the connector is running.

The Mainframe connector is designed for TN3270 applications and built on the IBM Host Access API libraries. You must have the IBM Host Access API libraries before working with this connector. You can purchase these libraries from IBM.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Mainframe connector uses the following connection attributes:

Table 53—Mainframe Connector - Configuration parameters

| Parameters | Descriptions |
|------------------------------|---|
| host | The host of the server to which you are connecting. |
| port | The port the server is listening through. |
| user | The valid user name with which to connect to the server. |
| password | The password associated with the connection user. |
| logonRule | The rule used to log on to the application |
| logoffRule | The rule called to log off of the application |
| userIterateRegularExpression | The regular expression that should be used when fetching/iterating accounts. This expression breaks the screens into records that can be manipulated by the script. |
| userTransformRule | The rule called for each record delineated by the regular expression. This rule takes the text from the screens and converts it to a ResourceObject . |
| userIterateCommand | The command used to natively iterate over all users |
| defaultTimeout | The length of time scripts should wait for data to be returned during command execution. |

Schema attributes

Table 53—Mainframe Connector - Configuration parameters

| Parameters | Descriptions |
|--------------------|--|
| defaultIdleTimeout | The length of time the screen should be idle before timing out. |
| morePrompt | The prompt scripts should expect to receive to indicate there is more data on the screen |
| readyPrompt | The prompt scripts should expect to receive to indicate the mainframe is ready |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 54—Mainframe Connector - Account Attributes

| Name | Description |
|----------------|---|
| USER | The user ID or login ID of the user. |
| NAME | The user's name. |
| DEFAULT-GROUP | The default group to which the owner of the attribute belongs. |
| OWNER | The owner of the profile, or object. |
| SECURITY-LABEL | The security label assigned to the data being collected as defined by the Open Systems Interconnection Security Architecture. |
| ATTRIBUTES | The attributes assigned to the user. |
| GROUP | Group ID for the owner group. |

Chapter 47: SailPoint IdentityIQ Novell Identity Manager Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 335 |
| Configuration parameters..... | 335 |
| Schema attributes | 336 |

Overview

The Novell Identity Manager connector uses the **groupMemberSearchDN** attribute as the starting point in the directory to start searching for ALL group memberships. The Novell Identity Manager does not store a user's group references on the user so this connector must always do a separate query to return a list of all of the user's groups.

Novell IDM connector can be used as both a multiplexing and a non-multiplexing connector. In the multiplexed mode, both aggregation and remediation happen through the IDM vault. In the non-multiplexed mode, aggregation happens through individual connectors but the removal and disabling of the account happens through the vault.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

This connector uses the following connection attributes:

Table 55—Novell Identity Manager Connector - Configuration parameters

| Attribute | Description |
|---------------------|--|
| isMultiplexing | Specifies if this connector is multiplexing. |
| useSSL | Specifies if the connection is over ssl. |
| authorizationType | Translates to the Context.SECURITY_AUTHENTICATION property in the api. The attribute value is one of the following strings: none , simple , strong . |
| user | The user to connect as. Typically a DN string such as Administrator. |
| password | The password for the administrator account. |
| port | The port the server is listening through. |
| host | The host of the server. |
| searchScope | The depth to search the LDAP tree. OBJECT_SCOPE , ONELEVEL_SCOPE , and SUBTREE_SCOPE . |
| searchDN | The search starting point. This is a DN string. |
| iterateSearchFilter | An optional filter that can be added to the configuration to scope the objects returned when the iterateObjects method is called. |

Table 55—Novell Identity Manager Connector - Configuration parameters (Continued)

| Attribute | Description |
|----------------------|--|
| pageSize | The number of objects to get, per page, when iterating over large numbers of objects. The default is 500. |
| groupMemberSearchDN | Where to start in the tree when resolving a user's group membership. This is a DN String. |
| filterString | This setting can be used to filter object as they are returned for an underlying application. Derived attributes can also be included in the filter. |
| groupMemberAttribute | The name of the attribute used on the server to store group members. |

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 56—Novell Identity Manager Connector - Account Attributes

| Name | Description |
|------------------|--|
| businessCategory | The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: "engineering", "finance", and "sales". |
| carLicense | This attribute type contains the license plate or vehicle registration number associated with the user. |
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| dn | This attribute contains the distinguished name by which the user is known. |
| departmentNumber | This attribute contains a numerical designation for a department within your enterprise. |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Table 56—Novell Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|--------------------------|--|
| destinationIndicator | <p>This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom.</p> <p>Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.</p> |
| displayName | This attribute contains the preferred name to be used for this person throughout the application. |
| employeeNumber | This attribute contains the numerical identification key for this person within your enterprise. |
| employeeType | This attribute contains a descriptive type for this user, for example, contractor, full time, or part time. |
| facsimileTelephoneNumber | This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute. |
| givenName | <p>This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute.</p> <p>Examples: "John", "Sue", and "David".</p> |
| groups | <p>This attribute type contains a list of groups of which this person is a member.</p> <p>Example: "Sales" or "Engineering"</p> |
| homePhone | This attribute contains the employee's home phone number. |
| homePostalAddress | This attribute contains the employee's mailing address. |
| initials | <p>This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute.</p> <p>Examples: "J. A." and "J".</p> |
| internationalISDNNumber | <p>This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute.</p> <p>Example: "0198 444 444".</p> |
| l | <p>This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute.</p> <p>Examples: "Austin", "Chicago", and "Brisbane".</p> |
| mail | This attribute type contains the RFC822 mailbox for the user. |

Table 56—Novell Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|----------------------------|---|
| manager | This attribute type contains the distinguished name of the manager to whom this person reports. |
| mobile | This attribute type contains the mobile telephone number of this person. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| pager | This attribute type contains the telephone number of this persons pager. |
| physicalDeliveryOfficeName | This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E". |
| postOfficeBox | This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27". |
| postalAddress | This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA". |
| postalCode | This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA. |
| preferredDeliveryMethod | This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone". |
| preferredLanguage | This attribute type contains the preferred written or spoken language of this person. |
| registeredAddress | This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA". |
| roomNumber | This attribute type contains the room or office number or this persons normal work location. |
| secretary | This attribute type contains the distinguished name of this persons secretary. |

Table 56—Novell Identity Manager Connector - Account Attributes (Continued)

| Name | Description |
|---------------------------|---|
| seeAlso | This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object “cn=Elvis Presley,ou=employee,o=xyz\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.”. Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values. |
| sn | This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: “Smith”. |
| st | This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”. |
| street | This attribute type contains site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”. |
| telephoneNumber | This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. |
| teletexTerminalIdentifier | The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute. |
| telexNumber | This attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute |
| title | This attribute type contains the persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”. |
| uid | This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”. |
| objectClass | The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 57—Novell Identity Manager Connector - Group Attributes

| Name | Description |
|--------------|---|
| cn | This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12". |
| uniqueMember | This attribute type contains the groups to which this person is a unique member. |
| dn | This attribute type contains the directory path to the object. |
| o | This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated." |
| ou | This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies". |
| owner | This attribute type contains the distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is "cn=All Employees, ou=Mailing List,o=xyz, Inc.", is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): "cn=Human Resources Director,ou=employee,o=xyz, Inc." |
| description | This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales". |

Chapter 48: SailPoint IdentityIQ RACF Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 341 |
| Configuration parameters..... | 341 |
| Schema Attributes..... | 342 |
| Account attributes..... | 342 |
| Group attributes..... | 346 |

Overview

The RACF connector was developed to read the file produced by the RACF unload utility.

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The RACF connector uses the following configuration parameters:

Table 58—RACF Connector - Configuration parameters

| Parameters | Description |
|-------------------------|--|
| filetransport | local, ftp, scp |
| host | The host of the server to which you are connecting. |
| transportUser | The user to use with ftp and scp. Not valid with local. |
| transportUserPassword | The password to use with of ftp and scp. Not valid with local. |
| file | The fully qualified path to the file. |
| fileEncoding | Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: http://www.iana.org/assignments/character-sets If this field is empty, the default encoding (the value of file.encoding specified by the jvm) is used. |
| mapToResourceObjectRule | Rule that is called to override the transformation of the data from the Map<String,String> form into a ResourceObject . |
| filterString | Filter lines that match this string. |
| filterEmptyRecords | If activated, records that have no data are filtered. |

Table 58—RACF Connector - Configuration parameters (Continued)

| Parameters | Description |
|-----------------------------------|---|
| preIterativeRule | <p>The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values.</p> <p>This rule is called after the file has been transferred, but before iteration over the objects in the file is started.</p> <p>For validation this rule can use the existing statistics stored by the postIterationRule during the last aggregation. The rule can compare the stored values with the new values to check for problems</p> |
| postIterativeRule | <p>The post-iterate rule can store away the configuration object and rename/delete the file if desired.</p> <p>This rule is called after aggregation has completed and ALL objects have been iterated.</p> |
| RACF Attribute Customization Rule | <p>The rule used to extend the parsing capabilities to customer records or redefine existing record configurations.</p> <p>The RACF attribute customization rule creates a map of LineRecord objects that hold the record ID and other field definitions.</p> |

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 59—RACF Connector - Account Attributes

| Attribute | Description |
|-------------------|---|
| CLASSES | |
| CATEGORIES | Defines the categories associated with a general resource. There is one record per general resource/category combination. |
| KERB_NAME | RACF user name as taken from the profile. |
| KERB_MAXLIFE | Maximum ticket life. |
| KERB_KEY_VER | Current key version. |
| KERB_ENCRYPT_DES | Is key encryption using DES enabled? |
| KERB_ENCRYPT_DES3 | Is key encryption using DES3 enabled? |
| KERB_ENCRYPT_DESD | Is key encryption using DES with derivation enabled? |

Table 59—RACF Connector - Account Attributes (Continued)

| Attribute | Description |
|-------------------|--|
| KERB_ENCRYPT_A128 | Is key encryption using AES128 enabled? |
| KERB_ENCRYPT_A256 | Is key encryption using AES256 enabled? |
| KERB_KEY_FROM | Key source. Valid values are PASSWORD or PHRASE. |
| NAME | User ID as taken from the profile name. |
| CREATE_DATE | The date that the profile was created. |
| OWNER_ID | The user ID or group name that owns the profile. |
| ADSP | Does the user have the ADSP attribute? |
| SPECIAL | Does the user have the SPECIAL attribute? |
| OPER | Does the user have the OPERATIONS attribute? |
| REVOKE | Is the user REVOKEd? |
| GRPACC | Does the user have the GRPACC attribute? |
| PWD_INTERVAL | The number of days that the user's password can be used. |
| PWD_DATE | The date that the password was last changed. |
| PROGRAMMER | The name associated with the user ID. |
| DEFGRP_ID | The default group associated with the user. |
| LASTJOB_TIME | The time that the user last entered the system. |
| LASTJOB_DATE | The date that the user last entered the system. |
| INSTALL_DATA | Installation-defined data. |
| UAUDIT | Do all RACHECK and RACDEF SVCs cause logging? |
| AUDITOR | Specifies if the user has the auditor attribute. |
| NOPWD | YES - indicates that this user ID can logon without a password using OID card. NO - indicates that this user must specify a password. PRO - indicates a protected user ID. PHR - indicates that the user has a password phrase. |
| OIDCARD | Specifies if this user has the OIDCARD data. |
| PWD_GEN | The current password generation number. |
| REVOKE_CNT | The number of unsuccessful logon attempts. |
| MODEL | The data set model profile name. |
| SECLEVEL | The user's security level. |
| REVOKE_DATE | The date that the user will be revoked. |
| RESUME_DATE | The date that the user will be resumed. |
| ACCESS_SUN | Can the user access the system on Sunday? |
| ACCESS_MON | Can the user access the system on Monday? |

Table 59—RACF Connector - Account Attributes (Continued)

| Attribute | Description |
|--------------------|---|
| ACCESS_TUE | Can the user access the system on Tuesday? |
| ACCESS_WED | Can the user access the system on Wednesday? |
| ACCESS_THU | Can the user access the system on Thursday? |
| ACCESS_FRI | Can the user access the system on Friday? |
| ACCESS_SAT | Can the user access the system on Saturday? |
| START_TIME | After what time can the user logon? |
| END_TIME | After what time can the user not logon? |
| SEC_LABELS | The user's default security label. |
| ATTRIBS | Other user attributes (RSTD for users with RESTRICTED attribute). |
| PWDENV_EXISTS | Has a PKCS#7 envelope been created for the user's current password? |
| PWD_ASIS | Should the password be evaluated in the case entered? |
| PHR_DATE | The date the password phrase was last changed. |
| PHR_GEN | The current password phrase generation number. |
| CERT_SEQN | Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered. |
| PPHENV_EXISTS | Has the user's current password phrase been PKCS#7 enveloped for possible retrieval? |
| ASSOCIATED_MAPPING | Defines the certificate name filter in the DIGTNMAP class associated with this user ID. |
| CSDATA_CUSTOM | Record type of the User CICS Data record |
| LNOTES_SHORTNAME | User ID as taken from the profile name. |
| CICS_OP_CLASSES | The class associated with the CICS operator. |
| GROUPS | |
| OVM_UID | User identifier (UID) associated with the user name from the profile. |
| OVM_HOME_PATH | Home path associated with the user identifier (UID). |
| OVM_PROGRAM | Default program associated with the user identifier (UID). |
| OVM_FSROOT | File system root for this user. |
| PRIMARY_LANGUAGE | The primary language for the user. |
| SECONDARY_LANGUAGE | The secondary language for the user. |
| CICS_RSL_KEY | Defines the resource security level (RSL) keys associated with a CICS user. There is one record per combination of user and CICS RSL key. |
| LDAP_HOST | LDAP server URL. |

Table 59—RACF Connector - Account Attributes (Continued)

| Attribute | Description |
|----------------------|--|
| LDAP_BIND_DN | LDAP BIND distinguished name. |
| NETVIEW_IC | Command list processed at logon. |
| NETVIEW_CONSOLE_NAME | Default console name. |
| NETVIEW_CTL | CTL value: GENERAL, GLOBAL, or SPECIFIC. |
| NETVIEW_MSGRECV | Eligible to receive unsolicited messages? |
| NETVIEW_NGMFADMN | Authorized to NetView graphic monitoring facility? |
| NETVIEW_NGMFVSPN | Value of view span options. |
| NDS_UNAME | NDS user name associated with the user ID. |
| CICS_OPIDENT | The CICS operator identifier. |
| CICS_OPPTY | The CICS operator priority. |
| CICS_NOFORCE | Is the extended recovery facility (XRF) NOFORCE option in effect? |
| CICS_TIMEOUT | The terminal time-out value. Expressed in hh:mm. |
| DCE_UUID | DCE UUID associated with the user name from the profile. |
| DCE_NAME | DCE principal name associated with this user. |
| DCE_HOMECCELL | Home cell name. |
| DCE_HOMEUUID | Home cell UUID. |
| DCE_AUTOLOGIN | Is this user eligible for an automatic DCE login? |
| CERTIFICATE | Defines the names of the certificate profiles in the DIGTCERT class that are associated with this user ID. |
| CICS_TSL_KEY | Defines the transaction security level (TSL) keys for a CICS user. There is one record per combination of user and CICS TSL key. |
| TSO_ACCOUNT_NAME | User ID as taken from the profile name. |
| TSO_COMMAND | The command issued at LOGON. |
| TSO_DEST | The default destination identifier. |
| TSO_HOLD_CLASS | The default hold class. |
| TSO_JOB_CLASS | The default job class. |
| TSO_LOGIN_PROC | The default logon procedure. |
| TSO_LOGIN_SIZE | The default logon region size. |
| TSO_MSG_CLASS | The default message class. |
| TSO_LOGON_MAX | The maximum logon region size. |
| TSO_PERF_GROUP | The performance group associated with the user. |
| TSO_SYSOUT_CLASS | The default sysout class. |
| TSO_USER_DATA | The TSO user data, in hexadecimal in the form X<cccc>. |

Schema Attributes

Table 59—RACF Connector - Account Attributes (Continued)

| Attribute | Description |
|---------------------|---|
| TSO_UNIT_NAME | The default SYSDA device. |
| TSO_SECLABEL | The default logon security label. |
| DFP_DATA_RECORDS | Defines the information required by the System Managed Storage facility of the Data Facility Product (DFP). |
| AREA_NAME | Area for delivery for the user. |
| BUILDING | Building for delivery. |
| DEPARTMENT | Department for delivery. |
| ROOM | Room for delivery. |
| ADDRESS1 | Address line 1. |
| ADDRESS2 | Address line 2. |
| ADDRESS3 | Address line 3. |
| ADDRESS4 | Address line 4. |
| ACCOUNT_NUMBER | User account number for delivery. |
| MVS_UID | z/OS UNIX user identifier (UID) associated with the user name from the profile. |
| MVS_HOME_PATH | HOME PATH associated with the z/OS UNIX user identifier (UID). |
| MVS_PROGRAM | Default Program associated with the z/OS UNIX user identifier (UID). |
| MVS_MAX_CPU TIME | Maximum CPU time associated with the UID. |
| MVS_MAX_ASSIZE | Maximum address space size associated with the UID. |
| MVS_MAX_FILEPROC | Maximum active or open files associated with the UID. |
| MVS_MAX_PROC | Maximum number of processes associated with the UID. |
| MVS_MAX_THREADS | Maximum number of threads associated with the UID. |
| MVS_MAX_MAP_STORAGE | Maximum mappable storage amount associated with the UID. |
| MVS_MEM_LIMIT | Maximum size of non-shared memory. |
| MVS_SHMEM_LIMIT | Maximum size of shared memory. |
| NETVIEW_OPCLASS | OPCLASS value from 1 to 2040. |
| EIM_LDAPPROFILE | EIM LDAPBIND profile name. |

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 60—RACF Connector - Group Attributes

| Attribute | Description |
|----------------|--|
| SUBGROUPNAME | The name of a subgroup within the group. |
| MVS_GID | OMVS z/OS UNIX group identifier (GID) associated with the group name from the profile. |
| CSDATA_CUSTOM | Defines the custom fields associated with a group. There is one record per combination of group and CSDATA custom fields. |
| MEMBERS | A user ID within the group. |
| NAME | Group name as taken from the profile name. |
| SUPERIOR_GROUP | Name of the superior group to this group. |
| CREATE_DATE | Date that the group was defined. |
| OWNER_ID | The user ID or group name which owns the profile. |
| UACC | The default universal access. Valid values are NONE for all groups other than the IBM-defined VSAMDSET group which has CREATE. |
| NOTERMUACC | Indicates if the group must be specifically authorized to use a particular terminal through the use of the PERMIT command. |
| INSTALL_DATA | Installation-defined data. |
| GROUP_MODEL | Data set profile that is used as a model for this group. |
| UNIVERSAL | Indicates if the group has the UNIVERSAL attribute. |
| OVM_GID | OMVS z/OS UNIX group identifier (GID) associated with the group name from the profile. |
| TME_ROLE | Role profile name. |

Chapter 49: SailPoint IdentityIQ Rule Based Logical Connector

The following topics are discussed in this chapter:

| | |
|-------------------------------|-----|
| Overview..... | 349 |
| Configuration parameters..... | 349 |

Overview

The Rule Based Logical connector was developed to create objects that function like applications in the IdentityIQ product, but that are actually formed based on the detection of accounts from other applications in existing identity cubes.

For example, you might have one logical application that represents three other accounts on different applications, an Oracle database, an LDAP authorization application, and a custom application for internal authentication. The logical application rule scans identities and creates an account on the logical application each time it detects the three required accounts on a single identity.

You can then use the single, representative account instead of the three separate accounts from which it is comprised for certification, reporting, and monitoring throughout the product.

Configuration parameters

The Rule Based Logical connector uses the following rules to assign accounts:

Table 61—Rule Based Logical Connector - Rules

| Parameters | Description |
|--------------------------|--|
| Logical Application Rule | Enter the name of a logical application rule. The logical application rule defines the requirements that must be met before an identity is assigned an account on this logical application. |
| Logical Remediation Rule | Enter the name of a logical remediation rule. The logical remediation rule defines how remediation request for the logical application account or any of the accounts with which it is comprised are handled. |

Configuration parameters

Appendix Section

This section contains the information on the following:

- A: Password Interceptor on page 353
- B: IQService Before/After Script on page 373
- C: Delta Aggregation on page 379

Appendix A: Password Interceptor

This appendix describes the following information.

| | |
|--|-----|
| Password Interceptor on IdentityIQ Server | 353 |
| Password Interceptor for Microsoft Active Directory. | 354 |
| Installation | 354 |
| Uninstallation | 358 |
| Password Interceptor for LDAP. | 358 |
| Installation | 358 |
| Uninstallation | 362 |
| Password Interceptor for UNIX. | 363 |
| (Only for AIX) Using aix-pwi-set.sh script | 363 |
| Installation | 364 |
| Verifying the Password Interceptor installation | 367 |
| Uninstallation | 370 |
| Managing Password Interceptor messages | 370 |
| Manually Start/Stop Password Interceptor Client. | 371 |
| Troubleshooting | 372 |

Password Interceptor on IdentityIQ Server

The password interceptor functionality on the IdentityIQ server is managed through a web service method and a workflow. The password interceptor client calls the web service which in turn launches the workflow to complete the password interception process (usually propagation to other systems).

Password Intercept Web Service

The password interceptor client invokes the IdentityIQ password intercept web service on the IdentityIQ server, passing it a JSON string containing the application name (as recorded in IdentityIQ), the user's nativeIdentity on the application, and the new password. The web service identifies the link associated with that application/nativeIdentity and thereby locates the Identity to which it applies. It then launches a workflow to process the required password interception functionality, passing it the Identity name, application name, nativeIdentity value, and encrypted password.

Password Intercept Workflow

The workflow to launch is designated as the workflowPasswordIntercept in the IdentityIQ System Configuration attributes map (for example, <entry key="workflowPasswordIntercept" value="Password Intercept">). By default, this points to the **Password Intercept** workflow that ships with IdentityIQ; this workflow can be used as written or customized, or the **workflowPasswordIntercept** attribute can be pointed to a custom workflow if required to meet business needs.

The default **Password Intercept** workflow (Password Intercept) is designed to sync the intercepted password to other accounts held by the Identity. It contains two process variables that can be used to determine which application account passwords should be synced. By default, the **syncAll** variable is set to true, which means the workflow will sync the password to all accounts held by the Identity on all applications. To limit the password sync to a subset of applications, edit the workflow, setting **syncAll** to false and specifying the application names (as a CSV or a List of strings) to which the password should be synced in the **targetApplications** variable. The workflow creates a provisioning policy to push the password changes to the target applications.

When other conditions exist for identifying the target applications for password sync, they can be specified in the Password Intercept workflow's Select Targets step; this step is empty by default and exists only as a placeholder to facilitate this customization. If the business wants password interception to drive a completely different set of actions, the workflow's steps can be customized to meet those needs or a different custom workflow can be specified for the installation's password intercept functionality.

Password Interceptor for Microsoft Active Directory

This section describes the installation and configuration of IdentityIQ Active Directory Password Interceptor Client Service.

Password Interceptor for Active Directory provides the mechanism by which a password change initiated by an Active Directory user is captured by the Client and sent to IdentityIQ.

With Active Directory, there can be multiple domain controllers in a particular domain and multiple Domains in the Domain Tree. The Password Interceptor captures a password change from any Domain Controller in the Active Directory Domain Tree.

The Password Interceptor Client service intercepts each password change and sends it to the Password Interceptor Server configured in IdentityIQ. The Client service must be installed on each Domain Controller.

The following points describe the sequence of events triggered when an Active Directory user changes password:

1. A user requests a Password change on a workstation which belongs to one Domain Controller (DC) in the Domain.
2. On the DC, the Local Security Authority (LSA) calls the password filter that is registered on the computer. Password filters provide a means to implement password policy and change notification. When a user makes a password change request on a Domain Controller, the Local Security Authority (LSA) calls the password filters registered on the system.

For more information on the Password filter for Microsoft Windows, see [Using Password Filters](#).

3. The Password Change request is written to the Password Interceptor Client which is a Windows service running on the same computer.
4. The intercepted messages containing the password data from the Password Interceptor Client are sent to the IdentityIQ Server.

The connection details of the IdentityIQ Server, including the IdentityIQ host name and port are stored in an Active Directory object of **serviceConnectionPoint** class. It is stored in Organizational Unit object and named as **PWDINT_SERVER_OU**.

In order to get each and every password change event detail, Password Interceptor Client must be installed on each Domain Controller present in the domain.

Installation

This section describes the installation procedure of Password Interceptor for Active Directory.

Hardware and Software requirement

The following table lists the required hardware and software requirements:

| Component | Description |
|------------------|---|
| Operating System | Microsoft Windows Server 2003 Microsoft Windows Server 2003 R2 Microsoft Windows Server 2008 Microsoft Windows Server 2008 R2 (x86 or x64) |
| Computer | Any hardware configuration supported by any of the Microsoft Active Directory operating systems. |
| Memory | 32 MB RAM or greater |
| Disk Space | 5 MB or greater |

Installation parameters

[Table 62—Installation parameters](#) describes the various parameter required to install and configure Password Interceptor. The first installation is the initial installation of Password Interceptor Client in the domain which creates the **serviceConnectionPoint** object. This will be referred by all clients installed in the domain.

Table 62—Installation parameters

| Command line | Description | First installation | Addition installation |
|--------------------------------|--|--------------------|-----------------------|
| -i | This parameter used to install password interceptor service | Required | Required |
| -first | This parameter indicates whether it is first installation on password interceptor service. | Required | Not required |
| -oupath <path of OU> | This parameter accepts Active Directory path for creating the serviceConnectionPoint object, which saves Password Interceptor configurations and IdentityIQ server location. During first time installation this OU will be created. For all other subsequent installation already created (that is, OU which is created in the first installation) OU path must be provided. Syntax: -oupath <distinguished name of location> | Required | Required |
| -server <IIQ hostname or IP> | This parameter accepts hostname or IP of IdentityIQ server. | Required | Not required |
| -port <port number> | This parameter accepts port number used for communication. | Required | Not required |
| -user <iiq admin username > | This parameter takes IdentityIQ administartor Username. This option should be used along with -password option. | Required | Not required |
| -password <iiq admin password> | This parameter takes IdentityIQ administrator Password. This option should be used along with -user option. | Required | Not required |

Table 62—Installation parameters

| Command line | Description | First installation | Addition installation |
|--|---|--------------------|-----------------------|
| -application <name of AD connector application on IdentityIQ > | This parameter accepts name of the IdentityIQ Actice Directory application. | Required | Not required |
| -trustAll <Y/N> | Whether to trust all SSL certificates or not. Not applicable if secure option is set to N. Default : N | Optional | Optional |
| -secure <Y/N> | Whether to use secured protocol for communication between Client service and IdentityIQ server. Default : Y | Optional | Optional |
| -interception_time out <time in hours> | The length of time (in hours) for the Password Interceptor Client to retain a password change record that could not be sent to the Password Server. Once this interval has elapsed, the record is discarded by the Password Interceptor Client and the password change will not be sent to IdentityIQ. Default : 24 Example: -interception_timeout "24" | Optional | Optional |
| -retry_server_connect <time in seconds> | The interval (in seconds) at which the Password Interceptor Client attempts to Communicate with the Password Server. Default : 300 Example: -retry_server_connect "300" | Optional | Optional |
| -l <0-3> | Log level. Default : 0 This option can not be used in combination with -i option. Example: -l "1" | Optional | Optional |
| -f <log file path> | Example: -f "C:\Program Files\IdentityIQ \ADPwdClient" This option can not be used in combination with -i option. Default: Password Interceptor Client Home folder | Optional | Optional |
| -iiq_response_time out <time in seconds> | Time in seconds to wait for response from IdentityIQ. If this parameter is not set then the default value is 25 seconds. | Optional | Optional |
| -searchfilter <AD searchfilter string> | Active Directory searchfilter string to filter object types. | Optional | Optional |
| -scope <OU1 DN [+] OU2 DN...[+]...[+]...> | List of OU path to limit scope of interceptions. Multiple OU distinguished names are separated by this set of character- "[+]". Example: -scope "TESTOU1 DN [+] TESTOU2 DN [+] TESTOU3 DN" where DN= Distinguishe Name of container. | Optional | Optional |
| -url <iiq server url> | URLof IdentityIQ server. Directly takes URL instead of -server and -port . If URL is porvided then -server , -port and -secure values will be ignored. | Optional | Optional |

Installation process

1. Extract **ADPwdClient-6.0.zip** file.
2. Run **PwdClient.exe** with parameters as explained in [Table 62— Installation parameters](#)

For Example:

- a. For first installation:

```
PwdClient.exe -i -first -oupath "DC=ABC,DC=DEF" -server "117.16.55.11" -port "8888"
-user"admin" -password "adminpassword" -application "AD" -trustall "n" -secure
"Y"-interception_timeout "26" -retry_server_connect "300"
```

This will create a Windows Service named IdentityIQ **Password Interceptor Client** and will copy the filter dll named **IIQPWDINT.dll** in system32 folder and will register the dll. Also it will create a OU named **PWDINT_SERVER_OU** in **DC=ABC,DC=DEF**. The ServiceConnectionPoint object name **HOST_ADDR_117.16.55.11** will be created within the **PWDINT_SERVER_OU**. This object will hold the connection details, application name and other details provided in the above command. After successful installation, the command will prompt the appropriate message and will recommend a reboot.

- b. For additional installation:

```
PwdClient.exe -i -oupath "DC=ABC,DC=DEF"
```

This will create a Windows Service named IdentityIQ **Password Interceptor Client** and will copy the filter dll named **IIQPWDINT.dll** in system32 folder and will register the dll. The OU path should match the path given in the first installation. The command will validate if the connection details exists at the given path. After successful installation, the command will prompt the appropriate message and will recommend a reboot.

3. For enabling secure communication, copy SSL certificate from IdentityIQ server and import it on the Client computer.
4. Reboot the system.

Managing Password Interceptor

This section describes how to manage and update Password Interceptor Client configuration properties.

Maintaining Client service

The following commands are used to manage Password Interceptor service:

Table 63—Commands for Password Interceptor

| Commands | Description |
|----------|------------------|
| -s | Start service |
| -k | Stop service |
| -t | Restart service |
| -r | Uninstall client |

For example: For restarting the service run the following command:

```
PwdClient.exe -t
```

Password Interceptor for LDAP

Updating Configuration Properties

To update any of the configuration parameter from table 3.1, run **PwdClient.exe** with that parameter and value as argument.

For example,

- To update log level:
PwdClient.exe -l "3"
- To update port number:
PwdClient.exe -port "8443"
- To update interception timeout
PwdClient.exe -interception_timeout "48"

Properties of Password Interceptor can be updated from any Password Interceptor service installed on any Domain Controller in the Domain. After updating any configuration setting, all Client services must be restarted.

Note: If you restart SailPoint Password Interceptor Client service, then all unsent password interceptions will be lost.

Uninstallation

Run the following command to uninstall the Password Interceptor Client:

PwdClient.exe -r

Reboot the system to complete uninstallation process.

Password Interceptor for LDAP

Whenever the password of a user changes, the password would be intercepted and sent across to IIQ over a secure socket and IdentityIQ in turn would synchronize the new password with applications configured for password change.

The SailPoint LDAP Password Interceptor (PWI) consists the following components:

- **LDAP PWI plugin:** The LDAP PWI plugin intercepts the password changes that occur on the directory server instance and keeps it in the memory.
- **Password Interceptor service:** The password interceptor service polls the PWI plugin for passwords intercepted by it and retrieves the changed passwords over SSL. It then relays the password change information to IdentityIQ server over SSL.
- **IdentityIQ server:** IdentityIQ server receives the password change and synchronizes to the applications configured for Password change.

Installation

This section provides the installation procedure of the LDAP Password Interceptor for SunOne.

Note: The steps mentioned in this section are applicable to all the platforms unless explicitly mentioned.

Supported platforms

The following operating systems are supported by Password Interceptor for LDAP (only for Sun One Directory Server 6.3):

- Microsoft Windows Server 2003 (32-bit only)
- Microsoft Windows Server 2008 (32-bit only)

Installing the LDAP PWI plugin

1. Login as the user under whose context you have installed the SunOne directory server instance. For example, Administrator.
2. Download the **LDAP_Plugin_<environment>.zip** file and extract it into a temporary folder.
3. In the extracted folder find the **vcredist_x86-8.0.61001.exe** and install it.

Note: The Redistributable Package installs runtime components of Visual C++ Libraries required to run applications developed with Visual C++ on a computer that does not have Visual C++ 2005 installed. This has been packaged since higher versions of this redistributable may not be meant for the Visual C++ 2005.

4. In the extracted folder, locate the **SAILPOINT_LOC** directory and copy it to the location where SunOne directory server instance has been installed.
5. Open the following files using a file editor and replace the **<SUNONE_INSTANCE_PATH>** text with the SunOne directory server instance path in the entire file except for the terminating path separator.

- <SunOne_Instance_path>/SAILPOINT_LOC/data/ Sailpoint_Plugins_install.Idif
- <SunOne_Instance_path>/SAILPOINT_LOC/data/ Sailpoint_Plugins.conf

Save the files and close.

6. Execute the `ldapmodify` command as follows:

```
> ldapmodify -h <hostname> -p <serverport> -D "<administrator>" -w  
"adminpassword" -a -f <SunOne_Instance_path>/SAILPOINT_LOC/data/  
Sailpoint Plugins install.ldif
```

On successful execution of the above command, the following messages are displayed:

adding new entry cn=SAILPOINTExtendedOp,cn=plugins,cn=config

adding new entry cn=SAILPOINTPostOp,cn=plugins,cn=config

Note: Enter the above command in the command prompt and do not copy and paste the above text.

Note: On Windows, the `ldapmodify` command can be found in `<SunOne Server Root>/dsrk6/bin` directory.

7. Enable the plugin logs for the directory server instance. On DSCC, select `serverinstance` and navigate to **serverConfiguration => Error Logging** tab and select the plugins checkbox.

8. Stop the server and if required, take a backup of the logs found at `<SunOne_Instance_path>/logs/*`. Delete all the current logs.

9. Start the SunOne directory server instance.

10. Open the error log specified in the **<SunOne Instance path>/logs/error** directory.

11. If the plugin logs are enabled and if the plugin install is successful, the following messages are displayed:

[illegible]

SAILPOINT_LOC: extOpInitializer - External operation plugin handler

successfully registered

SAILPOINT LOC: postOpInitializer - Post operation plugin handlers

successfully registered

[illegible]

Installing the Password Interceptor service on Windows

1. Login as the system administrator on the computer where the password interceptor service needs to be installed.
2. Download the **LDAPPWIClient.zip** file and extract into a temporary folder. From the extracted contents, copy the **SailpointPWIClient** folder to the same server where SunOne is installed or a different server.

Note: Java 1.5 or above has been installed.

3. If the password interceptor service is copied on a windows computer then perform the following steps:
 - a. Open the **SailpointPWIClient\Install.bat** file.
 - b. Locate the **JDK_HOME** string and specify the path of the JAVA_HOME directory upto the JAVA 'jre' folder. Do not include the 'jre' folder in the path.

Note: There should be no spaces in the JAVA_HOME path else the installation fails.

- c. Open a command prompt and navigate to the **SailpointPWIClient** directory and execute the **Install.bat** file.
- d. Open Windows Service (Start=> Run: services.msc). If the service is installed properly, the IdentityIQ **PasswordInterceptor** Service would be listed there. If it is not installed install the **Install.bat** file to determine whether the JDK_HOME path is specified properly.

For more information, see the README provided in the **SailpointPWIClient** directory.

Note: The **IdentityIQ PasswordInterceptor** service works only for one instance of the directory server and a single instance of the **IdentityIQ** server.

Note: The LDAP PWI plugin can work along with only one instance of **IdentityIQ PasswordInterceptor** service. The service should preferably be installed on the same computer as the SunOne directory instance.

4. Open the **SailpointPWIClient\config.cfg** file with a file editor.
5. Specify the parameters described in the following table. Any parameters that are not specified in the following table should not be changed for the successful operation of the IdentityIQ **PasswordInterceptor** service.

| Configuration parameter name | Description |
|------------------------------|--|
| ldaphost | Server name or IP address of machine where ldap server instance is running. |
| ldapport | LDAP server port. |
| ldapssl | Set this value to true if ldapport specified above is a SSL port. For a non SSL port, specify it as false . |
| ldapadmin | SunOne server instance administrator. Typically cn=Directory Manager |
| ldappwd | SunOne server instance administrator password. |
| iiqurl | IdentityIQ url. For example, <a href="http(s)://<hostname>:8080/identityiq">http(s)://<hostname>:8080/identityiq |
| iiqadmin | IdentityIQ administrator. |

| | |
|------------------------|--|
| iiqpwd | IdentityIQ administrator password. |
| ldap_specific_ous | This key is not present by default. Specify a list of containers or OUs separated by ' ' (pipe) character that contain users whose password change needs to be intercepted. Password change of any user other than the mapped OUs would not be sent to IdentityIQ. |
| ldap_pooling_namespace | This is the java namespace that gets looked up for providing java connection pooling mechanism. This is set to com.sun.jndi.ldap.connect.pool by default. |
| ldap_specific_ous | This key is not present by default. Specify a list of containers or OUs separated by ' ' (pipe) character that contain users whose password change needs to be intercepted. Password change of any user other than the mapped OUs would not be sent to IdentityIQ. |
| ldap_pooling_namespace | This is the java namespace that gets looked up for providing java connection pooling mechanism. This is set to com.sun.jndi.ldap.connect.pool by default. |
| iiqapps | This should be the application whose user has password changed. |
| pollinginterval | Polling time interval in milliseconds. |

Note: After executing the **IdentityIQ PasswordInterceptor** service, the credentials related information of the above file would automatically get encrypted. Post of which no data in the **config.cfg** file should be changed. Doing so, would result in invalid data being read by the service. You may start a fresh by backing up previous **config.cfg** and changing the above parameters to clear text. In this case, ensure that the **encryptfile** parameter is set to **true**.

6. Before you start the IdentityIQ **PasswordInterceptor** service, you should perform the SSL configuration between the:
 - IdentityIQ **PasswordInterceptor** and the SunOne LDAP server instance
 - IdentityIQ **PasswordInterceptor** and IdentityIQ.
7. Start the IdentityIQ **PasswordInterceptor** service.

Configuring SSL between 'IdentityIQ PasswordInterceptor' and the LDAP server

1. Open the SunOne directory server console also known as SunOne DSCC.
2. For the SunOne directory server instance, open the **Security Tab=> Certificate Tab** and click on the default **certificate=>ASCII** representation.
3. A text box appears showing the certificate in ASCII. Copy the entire contents including the **BEGIN CERTIFICATE** and **END CERTIFICATE** and paste the contents into a blank file.
4. Save the blank file as **defaultCert.cer** on the server where IdentityIQ **PasswordInterceptor** is installed.
5. On the server where the IdentityIQ **PasswordInterceptor** is installed, open a command prompt and run the keytool as follows :


```
>keytool -importcert -trustcacerts -alias defaultCert -file <location of defaultCert.cer> -keystore <JAVA_HOME>/jre/lib/security/cacerts
```
6. Perform the instructions as required to install the certificate file in the keystore.

Configuring SSL between 'IdentityIQ PasswordInterceptor' and IdentityIQ

Note: These steps are applicable if the IdentityIQ has been installed on Tomcat server

Password Interceptor for LDAP

1. Stop Tomcat server.
2. Edit the `<tomcat_home>/conf/Server.xml` file. Search for Connector tag with **scheme=https** and replace it with following:

```
<Connector port="8443" SSLEnabled="true"
protocol="org.apache.coyote.http11.Http11Protocol"
server="Apache" maxThreads="150" scheme="https" secure="true"
keystoreFile="/keystore/mykeystore.keystore" keystorePass="<password>"
clientAuth="false" sslProtocol="TLS" />
```

Note: The keystore file path (wrt tomcat_home) and password is specified.
3. Create a directory keystore in `<Tomcat_home>`.
4. To create the **mykeystore.keystore** keystore file, open a command prompt and navigate to `<Tomcat_home>/keystore` directory.
5. From the command prompt, run the keytool command as follows :

```
> keytool -keystore mykeystore.keystore -storetype jks -genkey -alias
myprivcert -keyalg RSA -keysize 2048
```
6. Provide the necessary details as required by the command and ensure that the name is same as the host name of the IdentityIQ Server and password as configured in [“Installing the LDAP PWI plugin” on page 359](#).
7. To export the certificate/public key, perform one of the following:
 - Access IdentityIQ from browser using https and save the certificate available in the browser tool bar
 - Or
 - Run the following command:

```
> keytool -exportcert -v -alias myprivcert -file test.cert -keystore mykeystore.keystore
```
8. Restart Tomcat.

Note: When a password is intercepted, it is cached in the LDAP server until it has been received by IdentityIQ. During its transit to IdentityIQ, stopping the directory server or the ‘IdentityIQ Password Interceptor’ service may loose the intercepted password. In this release, ‘IdentityIQ Password Interceptor’ service can be configured with a single application on IdentityIQ.

Uninstallation

This section describes the uninstallation procedure of LDAP PWI for SunOne on Windows.

1. To uninstall the IdentityIQ **PasswordInterceptor** service, perform the following:
 - a. Open a command prompt and navigate to the location where the service has been installed.
 - b. Execute the following command:
SailpointPWI.exe -uninstall IdentityIQ PasswordInterceptor
 - c. Manually delete the **LDAPPWIClient** folder.
2. To uninstall the LDAP PWI plugin:
 - a. Open a command prompt and execute the following command:

```
> ldapmodify -h <hostname> -p <serverport> -D "<administrator>" -w "adminpassword" -a -f
<SunOne_Instance_path>/SAILPOINT_LOC/data/ Sailpoint_Plugins_uninstall.ldif
```

- Note:** The `ldapmodify` command can be found in the `<SunOne Install folder>/dsrk6/bin` directory. Enter the above command in the command prompt and do not copy and paste the above text.
- Manually delete the **SAILPOINT_LOC** folder.
 - Restart the server.

Password Interceptor for UNIX

This section describes the considerations that relate to implementation of password Synchronization for the UNIX platform (AIX, Linux, and Solaris).

Password Interceptor is a process by UNIX Connector detects that an account has changed the password.

Installation of password interception for most managed systems does not require any additional action to implement password synchronization for accounts defined on the managed system.

(Only for AIX) Using aix-pwi-set.sh script

For a large number of accounts, implementing password synchronization for accounts defined on the managed system is unmanageable. The password interceptor installation contains the **aix-pwi-set.sh** script that facilitates large-scale implementation of password interception for AIX accounts.

The **aix-pwi-set.sh** script sets the required attribute for one or more AIX accounts to enable password interception for each account.

Enter the following command:

```
cd /etc/PWI
```

- To enable the password interception for all AIX accounts:
`./aix-pwi-set.sh -addall`
- To enable password interception for specific AIX accounts:
`./aix-pwi-set.sh -add account1 account2...`
- To disable password interception for all AIX accounts:
`./aix-pwi-set.sh -delall`
- To disable password interception for specific AIX accounts:
`./aix-pwi-set.sh -del account1 account2...`

Additional notes regarding **aix-pwi-set.sh** script: Enabling and disabling password interception for all accounts does not affect the root account.

Note: If new accounts are created on AIX computer after password interceptor installation, ensure that password interception is enabled for each new account in AIX using any of the methods described in this section above.

Note: Password Interceptor Client will only intercept password changes initiated by self and not by root user.

Installation

The Password Interceptor can be installed using the following procedures:

- Interactive installation
- Silent installation

Prerequisites

- Java Version 1.6 or later must be installed on the UNIX computer.
- JAVA_HOME and PATH variable should be set properly.
- Port number 6500 should be free which is used by **inetdcl**. To use a different port, modify **/etc/services** file manually.

Table 64— Installation parameters for Password Interceptor lists the details of the parameters used for Password Interceptor installation.

Table 64—Installation parameters for Password Interceptor

| Parameters | Description |
|---|--|
| INETDCL Directory | <p>The following prompt is displayed:</p> <p>Enter the directory where inetd client will be installed [/usr/sbin]:</p> <p>Directory where inetd client will be installed.</p> <p>Enter the directory in which the inetd client program will be installed. This directory is referred to as client-dir in the installation procedure. This directory must be on a local file system.</p> |
| PWI Client IP Address | <p>The following prompt is displayed:</p> <p>Select IP Address for PWI Client:</p> <p>IP address where password interceptor client is installed.</p> |
| PWI Client Port | <p>The following prompt is displayed:</p> <p>Select TCP/IP port number for PWI Client [6600]:</p> <p>Port number for password interceptor client.</p> |
| IIQ URL (IdentityIQ URL) | <p>The following prompt is displayed:</p> <p>- - Please specify the SailPoint IdentityIQ URL [http://localhost:8080/identityiq/]:</p> <p>Specify the URL for IdentityIQ.</p> |
| IIQ Application Name (IdentityIQ Application Name) | <p>The following prompt is displayed:</p> <p>Select SailPoint IdentityIQ Application Name for Password interception [operatingSystem]:</p> <p>Specify IdentityIQ Application Name configured here, <i>operatingSystem</i> is AIX, Linux, or Solaris.</p> |

Table 64—Installation parameters for Password Interceptor

| Parameters | Description |
|---|---|
| IIQ Administrator (IdentityIQ Administrator) | The following prompt is displayed: Select SailPoint IdentityIQ Admin [spadmin]: Specify the IdentityIQ administrator. |
| IIQ Administrator password (IdentityIQ Administrator password) | The following prompt is displayed: Select SailPoint IdentityIQ Admin Password: Specify IdentityIQ Administrator password. |
| INTERCEPTION_TIMEOUT (In minutes) | Used to set maximum time a password interception request will be held by PWIClient after which respective request will be destroyed. This parameter is not added in PWI.cfg file at the time of installation and is used with default values. It can be manually added to the PWI.cfg file to modify the default value. Default: 1440 min. |
| RETRY_SERVER_CONNECT (In seconds) | Used to set time interval after which PWIClient will retry connection to IdentityIQ server after failure. This parameter is not added in PWI.cfg file at the time of installation and is used with default values. It can be manually added to the PWI.cfg file to modify the default value. Default: 300 sec. |
| AIX PWDCK Directory | (Only for AIX) The following prompt is displayed: Enter the directory where AIX Password Check will be installed [/usr/sbin]: Directory in which to install the password interceptor exit module used to intercept passwords. |
| PWI Bit Type | (For Solaris and Linux) The following prompt is displayed: - Enter the required PWI Bit Type [32/64] [Default 32-bit]: |
| PAM Module Directory | (Only for Solaris)The following prompt is displayed: Enter the directory where Solaris PAM module will be installed [/usr/lib/security]: Directory where custom PAM module will be installed. This directory is referred to as PAM_LIB_DIR in the installation procedure. This directory must be on local file system & with write access to root user. |

Obtaining the Password Interceptor installation files

Extract the **SailPointIdentityIQPWIforoperatingSystemVersion.tar** tar file using the following command:

```
tar xvf SailPointIdentityIQPWIforoperatingSystemreleaseVersion.tar
```

where

- *operatingSystem* is AIX, Linux, or Solaris
- *releaseVersion* is the version of the current release

Interactive installation

This procedure is used to install a new instance of the Password Interceptor.

To perform the interactive installation for the Password Interceptor, perform the following:

1. Give executable permissions to **/tmp/AIXPWI/install_pwi.sh** using the following command:
`chmod 755 install_pwi.sh`
2. Specify the following command:
`/tmp/AIXPWI/install_pwi.sh`
3. The script displays a series of prompts, requesting data to customize the Password Interceptor. For information on responding to these prompts, see [Table 64— Installation parameters for Password Interceptor](#).
Several additional messages appear as customization continues. At the end of the installation procedure, the following message appears:
Installation ended successfully
4. To confirm the Password Interceptor installation, see [“Verifying the Password Interceptor installation” on page 367](#).

Silent installation

This procedure is used to perform a non-interactive installation of a new instance of the Password Interceptor from an installation image.

To perform the silent installation for the Password Interceptor, perform the following:

1. In a text editor open the **pwioperatingSystem.silent.properties** file and update the required values in the file and save the file.
For example, the following text lists the sample **pwiAix.silent.properties** file.

```
*****
      Silent installer property file
*****
#####
# PRODUCT NAME : SailPoint IdentityIQ Password Interceptor for AIX Version
6.1  #
#####
-----
# Directory where the Inetd client is installed
CLIENT_DIR=<Dir>
-----
# Interceptor Directory where aix_pwdck will be installed
AIX_PWDCK_DIR=<Dir>
-----
# Please specify PWI Client IP Address
PWI_HOST=<Hostname>

# Please specify PWI Client port
```

```

PWI_PORT=<Port>

# Please specify SailPoint IdentityIQ url here
# Default URL is http://localhost:8080/identityiq/rest/passwordIntercept
IIQ_URL=<url>
-----
# Please specify application name
APP=<Application Name>
-----
# Please specify SailPoint IdentityIQ admin name
IIQ_ADMIN=<Admin>
-----
# Please specify SailPoint IdentityIQ encrypted admin password
IIQ_ADMIN_PWD=<Admin Password>
-----
# Enable password interception for all users:
PWI_FOR_ALL_USRS=<Y/N>

```

The **`pwioperatingSystem.silent.properties`** file displays a series of prompts, requesting data to customize the Password Interceptor. For information on responding to these prompts, see [“Table 64—Installation parameters for Password Interceptor” on page 364](#).

2. Change the user context to a superuser or root.
3. Enter the following command to run the installation in silent mode:

```

/tmp/operatingSystemPWI/Install/install_pwi.sh -i silent -f
/tmp/operatingSystemPWI/Install/pwioperatingSystem.silent.properties

```
4. To confirm the Password Interceptor installation, see [“Verifying the Password Interceptor installation” on page 367](#).

Verifying the Password Interceptor installation

If the password interception support is installed, the following system files (**`/etc/services`** and **`/etc/inetd.conf`**) are updated and lines are added to them:

- (For AIX) **`/etc/services`** and **`/etc/inetd.conf`**
- (For Solaris) **`/etc/services`**, **`/etc/inet/inetd.conf`** and **`/etc/pam.conf`**
- (For Linux) **`/etc/services`**, **`/etc/xinetd.conf`** and **`/etc/pam.d/system-auth`**

The following steps describe the procedure for confirming that the system files have been updated correctly:

1. (For Solaris) The **`/etc/pam.conf`** system file must include a new entry that enables the Password Interceptor mechanism.
The Password Interceptor entry appears as follows:

```

password required bmc_pam_lib_path_name

```

For example,

```

# The next few lines were added by SailPoint Inc. for PWI
# SailPoint Inc. - item 1
other password required /usr/lib/security/libpam_sailpoint.so.1

```
2. The system file **`/etc/services`** must include a new entry that serves the Password Interceptor mechanism.

Password Interceptor for UNIX

A system service entry appears as follows:

- (For Linux and AIX) *servicePort/tcp*
- (For Solaris) *serviceName servicePort/tcp*

where *servicePort* is the tcp service port number and *serviceName* is the name of the service that needs to be used in **/etc/inet/inetd.conf** system file of Solaris.

For example,

```
# TCP/IP ports used by PWI of PM for AIX. Added by 6500-pwi -Line 01-
newd_pwi          6500/tcp                               # Added by 6500-pwi -Line 02-
```

where 6500 is the requested service port number.

3. Depending on the operating system perform one of the following:

- (For AIX) The **/etc/inetd.conf** system file must include a new entry that serves the Password Interceptor mechanism.

For example,

```
# PWI Modules of AIX Connector (newd_pwi)
newd_pwi stream tcp nowait root /usr/sbin/inetdcl /usr/sbin/inetdcl
```

A system service entry appears as follows:

```
service pwi stream tcp nowait root clientDir/inetdcl\
clientDir/inetdcl
```

where:

- newd_pwi: is the name of the service that is exactly used in **/etc/services** file.
- clientDir: is the directory on the local file system in which the inetd client program is installed.

- (For Linux) The **/etc/xinetd.conf** system file must include a new entry that serves the Password Interceptor mechanism.

For example,

```
# linux_con_pwi_start Dont remove, used for installation and
uninstallation purposes
```

```
# SA-Agent Modules (newd_pwi)
```

```
service newd_pwi
```

```
{
    socket_type      = stream
    protocol         = tcp
    wait             = no
    user             = root
    server           = /usr/sbin/inetdcl
    env              =
    instances        = UNLIMITED
}
```

```
#linux_con_pwi_end Dont remove, used for installation and uninstallation
purposes
```

A system service entry appears as follows:

```
service pwi stream tcp nowait root clientDir/inetdcl \
clientDir/inetdcl
```


where:

- `newd_pwi`: is the name of the service that is exactly used in `/etc/services` file.
- `clientDir`: is the directory on the local file system in which the `inetd` client program is installed.
- (For Solaris) The `/etc/inet/inetd.conf` system file must include a new entry that enables the Password Interceptor mechanism.

A system service entry appears as follows:

```
serviceName socket_type protocol wait-status user service_program
service_arguments
```

where `serviceName` is the name of a valid service listed in `/etc/services` file.

For example:

```
# PWI Modules of SailPoint IdentityIQ Password Interceptor for Solaris
Version 6.1 (newd_pwi)
newd_pwi stream tcp nowait root /usr/sbin/inetdcl /usr/sbin/inetdcl
```

4. (For Linux) The `/etc/pam.d/system-auth` system file must include a new entry that serves the Password Interceptor mechanism.

For example,

```
# The next line Must follow pam_cracklib.so and was added by SailPoint for
IdentityIQ password required /usr/lib/security/libpam_sailpoint.so
```

5. After successful installation, it will create `/etc/PWI` directory.

This directory contains the following files:

- **PWI.cfg**: configuration file for PWI containing all the fields specified during installation of PWI
- **SailPoint_PWI_install.log**: This log file is generated after PWI installation. This contains the log for PWI installation.
- **pwi.log**: This file is created after successful start of PWI Client. This will contain the log for PWI Client
- **PWIClient**: This directory will contain the JAR files related to PWI Client.
- **PWIClient/log4j.properties**: You can modify this file to enable logging for PWI client.
- Format of **PWI.cfg** file is as follows:

```
CLIENT_DIR=/usr/sbin
AIX_PWDCK_DIR=/usr/sbin
PWI_HOST=localhost
PWI_PORT=6600
IIQ_URL=http://localhost:8080/identityiq/
APP=AIX
IIQ_ADMIN=spadmin
```

IIQ_ADMIN_PWD=YWRtaW4=

- (For AIX only) **Aix-pwi-set.sh** and **aix-pwi**: This is used to enable password interception for user.

Uninstallation

The Password Interceptor can be uninstalled using the following procedures:

- Interactive uninstallation
- Silent uninstallation

Interactive uninstallation

To perform Interactive uninstallation for Password Interceptor, perform the following:

1. Log in to the required UNIX computer as user root.
2. Enter the following commands to give executable permissions to uninstall script:
`chmod 755 /etc/PWI/uninstall_pwi.sh`
3. Execute the following command:
`/etc/PWI/uninstall_pwi.sh`

Silent uninstallation

To perform silent uninstallation for Password Interceptor, perform the following:

1. Log in to the UNIX computer as user root.
2. Enter the following commands to give executable permissions to uninstall script:
`chmod 755 /etc/PWI/uninstall_pwi.sh`
3. Execute the following command:
`/etc/PWI/uninstall_pwi.sh -i silent`

Managing Password Interceptor messages

This section describes how to manage messages that are generated by the Password Interceptor. You can optionally determine whether or not to write password interceptor messages to the system logger, as well as the location to where the messages are written.

Note: If the **PWISyslog.conf** file does not exist, or if either of the parameters specified below are not present in the file, then Password messages are managed according the default values specified below.

To configure the **PWISyslog.conf** file, perform the following:

1. The **PWISyslog.conf** should be opened or created from **/etc/PWI/PWISyslog.conf**.
2. Insert or modify one or more entries using the following syntax:
parameter=value

The parameters and values that can be specified are described below:

- **PWI_WRITE_SYSLOG**: Whether to write Password Interceptor messages to the system logger. Possible values are Y and N. If set to N, the other parameters are ignored. Default: N
- **PWI_SYSLOG_LEVEL**: Destination of Password Interceptor messages (only applicable if **PWI_WRITE_SYSLOG** is set to Y). The value assigned to this parameter indicates which of the Priority parameters in **/etc/syslog.conf** should be used to determine where to write Password Interceptor messages.

Possible values for the **PWI_SYSLOG_LEVEL** parameter are:

| Value | Description |
|-----------|--|
| LOG_ERR | Messages are written to all the locations specified by parameters *.err, *.warn, *.notice, *.info and *.debug. |
| LOG_INFO | Messages are written to all the locations specified by parameters *.debug, and *.info |
| LOG_DEBUG | Messages are written to the location specified by parameter *.debug. Default. |

Note: Any line in the file that starts with # is regarded as a remark and is ignored. The **PWISyslog.conf** file should contain entries for both **PWI_WRITE_SYSLOG** and **PWI_SYSLOG_LEVEL**.

3. Save the file and exit.

Manually Start/Stop Password Interceptor Client

Password Interceptor Client is automatically started after installation of Password Interceptor. If required to start Password Interceptor Client manually, perform the following steps:

Start Password Interceptor Client

Execute the following command:

```
- java -jar /etc/PWI/PWIClient/pwiclient.jar > /dev/null 2>&1 &
```

Stop Password Interceptor Client

1. Verify if the Password Interceptor Client is running and get pid of Password Interceptor Client process.
2. Execute the following command:
 - `ps -eaf | grep 'pwiclient.jar' | grep -v 'grep' | awk '{print $2}'`
3. Kill this process by executing the following command:
 - `kill -9 <<pid>>`

Troubleshooting

1 - When you install password interceptor on Solaris zone computer with default paths for inetd client (/usr/sbin) and PAM module is to be installed (/usr/lib/security), the installation fails with an error.

When you install password interceptor on Solaris zone computer with default paths for inetd client (/usr/sbin) and PAM module is to be installed (/usr/lib/security), the installation fails with the following error message:

Installing Password Interceptor...

cp: cannot create /usr/lib/security/libpam_sailpoint.so.1: Read-only file system

Failed to install /etc/PWI/tmp/Files4d/Native/Solaris-32/libpam_sailpoint.so.1

Error - Aborting installation, Please uninstall PWI

Resolution: As for local zone, /usr/sbin and /usr/lib file systems are not accessible (as they belong to global zone), installation fails. Select the path with read/write access to local zone (like /opt etc) for installation of inetd client and PAM module.

2 - For Solaris Connector, Password Interception is successfully installed but interceptions are not sent to IdentityIQ.

For Solaris Connector, the following error message is displayed when Password Interception is successfully installed but interceptions are not sent to IdentityIQ:

Error: Error can be seen in /var/adm/message as

inetd[725]: [ID 702911 daemon.warning] Configuration file /etc/inet/inetd.conf has been modified since inetconv was last run. "inetconv -i /etc/inet/inetd.conf" must be run to apply any changes to the SMF

Resolution: Use the following command to restart **inetd** and then verify the interceptions:

svcadm restart inetd

3 - If you get the following error during installation of Password Interceptor

Bad: modifier in \$(-).

Resolution: Run the installer from the default shell of root user.Ex.sh

Appendix B: IQService Before/After Script

This appendix describes the following information.

| | |
|---|-----|
| Overview | 373 |
| Writing a script | 374 |
| Scripts with Object Oriented support | 374 |
| Scripts without Object Oriented support | 376 |
| Creating a Rule | 376 |
| Configuring the Rules in Application | 377 |

Overview

IdentityIQ provides most of the provisioning functionality for many systems through its connectors. Some systems provide better integration interface from Windows platform compared to other platforms. Hence connectors for such systems require IQService deployed on a Windows system. The IQService implementation performs the provisioning functions (such as Add User, Connect User to a Group) that are supported by the respective System. The IQService functions are called by the IdentityIQ connector implementation.

In addition to the basic action, some organizations may require supplementary actions performed by each function from Windows system. The IQService supports customization of the functions by allowing integrating before / after scripts implemented in any language. Following are some features of the IQService Before/After script:

- Centralized configurations (in IdentityIQ) for setting up Before/After scripts
- Supports Object Oriented scripting
- Script refers SailPoint library to get the request, result classes
- Can be executed with specific context
- Script can modify request/result

A script is a group of statements that perform one or more actions and manipulate request / result attributes. Scripts can be used to automate any required actions that are currently performed manually. Scripts called before processing the request are referred to as native before scripts and scripts called after processing the request are referred to as native after scripts.

The scripts needs to be defined in a Rule and then configured for an Application in IdentityIQ. Based on the rule type, IdentityIQ connector would send the scripts to IQService that needs to be executed before / after processing the request. The IQService supports executing before and after Rules for Create, Modify, and Delete request operations.

Writing a script

IQService divides scripts in the following categories:

- Scripts with Object Oriented support
- Scripts without Object Oriented support

Scripts with Object Oriented support

Scripting languages with Object Oriented capabilities (for example, PowerShell) are popular because of their simplistic nature and easy to use. These scripts can form objects of any type by referring any library/assembly implemented in any language and call its methods.

Native scripts implemented in these languages have easier and more powerful access to request and result objects. IQService comes with a class library named **Utils.dll** which bundles all required classes to access the request and result objects. The inputs provided to the script would be in the form of process environment variables. The following table describes the environment variables created by IQService:

| Name | Type | Before Script | After Script |
|-------------|--|---------------|--------------|
| Application | System.Collections.Hashtable | Read Only | Read Only |
| Request | SailPoint.Utils.objects.AccountRequest | Read/Write | Read Only |
| Result | SailPoint.Utils.objects.ServiceResult | Not Available | Read/Write |

The data in the environment variables is in XML. The script creates respective objects using **Utils.dll**. Once the object is modified, the script should convert it to XML by calling **toxml()** method of the object and write the xml to a file at the path that is passed as the only argument to the script. The script returns non-zero value in case of error and writes the error message in the file at the path that is passed as the argument to the script. This failure is communicated to IdentityIQ as part of result.

Sample PowerShell before script

Following is a sample PowerShell before script which modifies value of an attribute and add one new attribute to the request:

```
# Refer to SailPoint class library Requires PowerShell v2 installed on the
system.
Add-type -path utils.dll

# Read the environment variables
$reader = New-Object System.IO.StringReader ([System.String]$env:Request);

# Form the xml reader object
$xmlReader =
[System.Xml.XmlTextReader] ([SailPoint.Utils.xml.XmlUtil]::getReader($reader));

# Create SailPoint Request object
$requestObject = New-Object
SailPoint.Utils.objects.AccountRequest ($xmlReader);

# Loop through the attributes from the request
foreach ($attribute in $requestObject.AttributeRequests){
    if($attribute.Name -eq "description"){
```

```

$attribute.value = "my description";#change value of the attribute
}
}

# Add a new attribute to request
$attributeObject = New-Object SailPoint.Utills.objects.AttributeRequest;
$attributeObject.Name = "otherMobile";
$otherMobileValues = New-Object System.Collections.ArrayList;
$otherMobileValues.Add("222-292-2929");
$otherMobileValues.Add("333-292-2929");
$attributeObject.Value= $otherMobileValues;
$attributeObject.Operation = "Set";
$requestObject.AttributeRequests.Add($attributeObject);

# Write the request xml to file at the path passed as argument
$requestObject.toxml() | out-file $args[0];

```

Sample PowerShell after script

Following is a sample PowerShell after script which ensures that the request was processed successfully and creates home directory at the path specified in the request:

```

# Refer to SailPoint class library. Requires PowerShell v2 installed on the
system.
Add-type -path E:\SVN\trunk\src\WinRPCGateway\IQService\bin\Debug\utils.dll

# Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);
$sResult = New-Object System.IO.StringReader([System.String]$env:Result);

# Form the xml reader objects
$xmlReader = [
System.xml.XmlTextReader]([sailpoint.utills.xml.XmlUtil]::getReader($sReader
));
$xmlReader_Result = [
System.xml.XmlTextReader]([sailpoint.utills.xml.XmlUtil]::getReader($sResult
));

# Create SailPoint objects
$requestObject = New-Object
Sailpoint.Utills.objects.AccountRequest($xmlReader);
$resultObject = New-Object
Sailpoint.Utills.objects.ServiceResult($xmlReader_Result);

#Check if the request was processed successfully
if($responseObject.Errors.count -eq 0){

#Get Home directory path
    foreach ($attribute in $responseObject.AttributeRequests){
#Create Home directory
        if($attribute.Name -eq "TS_TerminalServicesHomeDirectory"){
            new-item $attribute.Value -itemtype directory;
        }
    }
}
}

```

Scripts without Object Oriented support

Non Object Oriented scripts do not support referring to the class library or a way of parsing XML. To have easy access to each attribute along with their operation and values, IQService creates process environment variables for each of the application and request attribute with name in the form **SP_<OPERATION>_<NAME>** for requests and **SP_APP_<NAME>** for application. For native identity, the environment variable would be **SP_NativeIdentity**. These types of scripts have limited access to result and get only **SUCCESS** or **FAIL** in the **Result** environment variable. Hence the after scripts implemented using these scripting languages cannot modify any attribute/result. The before scripts can add, modify, or remove any attribute from the request. The script needs to write the newly added or modified attribute to the file at the path passed as an argument to the script in the form **SP_<OPERATION>_<NAME>=<VALUE>**. For removing the attribute from the request, write **/~<ATTRIBUTE_NAME>** to the file. Value for the multivalued attribute is delimited by **/#**

Following is a sample batch after script which ensures that the request was processed successfully and creates home directory at the path specified in the request:

```
IF %Result% ==SUCCESS md %SP_Set_TS_TerminalServicesHomeDirectory%
```

Creating a Rule

IdentityIQ (6.0) user interface does not have facility to create Native Rule applicable for IQService. Create a rule with any supported type from the user interface. Add the script to the Rule source and save the Rule. Navigate to IdentityIQ debug, open the newly created Rule and perform following steps:

1. Change the rule type to one of the following types as appropriate:

| Type name | Description |
|-----------------------|--|
| ConnectorBeforeCreate | Before script for create operation. |
| ConnectorAfterCreate | After script for create operation. |
| ConnectorBeforeModify | After script for modify operation includes enable/disable, unlock. |
| ConnectorAfterModify | After script for modify operation includes enable/disable, unlock. |
| ConnectorBeforeDelete | Before script for delete operation. |
| ConnectorAfterDelete | After script for delete operation. |

2. Add the following attributes to the Rule in the form:

```
<Attributes>
  <Map>
    <entry key=<NAME> value=<VALUE>/>
  </Map>
</Attributes>
```

| Name | Description | Default Value |
|----------------------|---|---------------|
| ObjectOrientedScript | Whether the rule source uses object oriented scripting. | False |
| disabled | Set to true if the rule should not be executed on the IQService side. | False |

| Name | Description | Default Value |
|-----------|---|------------------------|
| extension | Extension of the script. | .bat |
| program | Program/application that can execute this type of script. Note: Ensure that this program is installed on the system where IQService is running and is properly configured to execute the scripts. | Cmd |
| timeout | Time interval (in seconds) for which IQService should wait for script to return. After this interval, IQService abort the script. | 10 |
| user | User name to run the script. | User running IQService |
| password | Password of the above user in encrypted form. Get the password encrypted using identityiq console using encrypt <PASSWORD> command. | |

Configuring the Rules in Application

With 6.0 releases, IdentityIQ user interface does not have facility to configure Native Rule applicable for IQService in Application. Navigate to IdentityIQ debug, open the application and add **<nativeRules>** under Attributes map with list of names of the Rules that must be configured for this application.

For example:

```
<entry key="nativeRules">
  <value>
    <List>
      <String>AfterCreate-Powershell</String>
      <String>BeforeCreate-Powershell</String>
      <String>BeforeModify-Batch</String>
    </List>
  </value>
</entry>
```


Appendix C: Delta Aggregation

This appendix describes the following information.

| | |
|---|-----|
| Overview | 379 |
| Delta aggregation for Microsoft Active Directory, ADAM, SunOne and Tivoli | 379 |
| Configuring server for Delta Aggregation | 380 |
| Testing Delta Aggregation | 380 |

Overview

Delta aggregation can be requested by checking a box on the task definition that is then passed through to the connector. If the connector does not support delta aggregation, then it ignores this flag and performs normal aggregation. The connectors supporting delta aggregation uses various mechanisms depending on the managed system to read the changes that have taken place after certain benchmark. It can be **lastModData**, **usnChanged**, or so on, else that indicates the last aggregation benchmark. This marker is stored on the application. Hence to take advantage of delta aggregation at least one full aggregation is required which will allow the connector to store the starting point for next delta aggregation.

If the volume of changes are more than 40% of the total data on the server, a normal aggregation run is recommended before over delta aggregation. The delta aggregation run can be scheduled at suitable interval considering the amount of changes happening on the managed system.

Delta aggregation for Microsoft Active Directory, ADAM, SunOne and Tivoli

1. Delta aggregation is supported for the following directory server types:

- Active Directory - Direct
- ADAM - Direct
- SunOne - Direct
- IBM Tivoli DS - Direct

Note: This includes changes such as user/group has been added/updated/deleted on the managed system. This version however does not aggregate delta changes for Move and Rename operations.

2. Prerequisite for delta aggregation:

After creating a fresh IdentityIQ 6.1 application of type ADAM - Direct, Active Directory - Direct, SunOne - Direct and IBM Tivoli DS - Direct or after an upgrade to IdentityIQ 6.1, open the application configuration file in debug mode and add the **GROUPS_HAVE_MEMBERS** feature string. Ensure that the following entries exist for respective application types:

- (For Active Directory - Direct): `<entry key='groupMemberAttribute' value='member'/>`

(For ADAM - Direct): `<entry key='groupMemberAttribute' value='member'/>`

For existing Active Directory - Direct or ADAM - Direct applications, add the following key into the applications configuration file. For new applications, modify the following key:

`<entry key="deletedObjectsContainer" value="CN=Deleted Objects,DOMAIN"/>`

Delta aggregation for Microsoft Active Directory, ADAM, SunOne and Tivoli

Where DOMAIN is a place holder for the naming context where the account and accountgroup objects reside. Replace **DOMAIN** with the corresponding naming context.

For example,

```
<entry key="deletedObjectsContainer" value="CN=Deleted Objects,dc=sailpoint,dc=com"/>
```

- (For SunOne - Direct and IBM Tivoli Directory Server - Direct):

```
<entry key='groupMemberAttribute' value='uniqueMembers'/>
```

Configuring server for Delta Aggregation

The mechanism used under the hood for Delta Aggregation is:

- (For AD/ADAM) uSNChanged
- (For Sun/Tivoli) changeLog

Following sections describe how to configure SunOne and Tivoli directory servers for delta Aggregation.

Note: After enabling the changelog on directory server, run Account and Account-Group full aggregation task before running delta aggregation.

Configuring SunOne directory server for Delta Aggregation

1. Locate the **dsconf** command of the SunOne directory server installation.
2. Using the command prompt execute the following command:
> dsconf set-server-prop --unsecured -h <host> -p <non ssl port> retro-cl-enabled:on
3. Enter the password for the directory server administrator.
4. Restart the server.

Configuring IBM Tivoli directory server for Delta Aggregation

1. Stop the Tivoli Directory server instance.
2. Locate the **idscfgchglg** for your Tivoli Directory Server installation.
3. To configure a change log for directory server instance, run the following command:
idscfgchglg -l <Tivoli instance> -m 0
4. Start the directory server instance.

Note: Confirm the server has been enabled for changelog, open a ldap browser and bind to the ldap server instance and view the cn=changelog naming context. You should be able to see this naming context and the relevant change objects. Ensure this before you proceed with delta aggregation for SunOne and Tivoli directory servers.

Testing Delta Aggregation

For delta aggregation to work properly, it needs a start point from where it would detect changes. To retrieve changes from the last iteration, it needs to first perform a full aggregation during which it maintains its reference point. Once the full aggregation completes, you may create a separate delta aggregation task to retrieve delta changes that occurred post the full aggregation.

Perform the following steps to test delta Aggregation:

1. Execute Account and Account - Group Aggregation task.

2. Create a task with delta aggregation flag set for Account and Account - Group Aggregation.
3. Perform Create/Update/Delete/Revoke operations for Accounts/Groups on the directory server.
4. Execute the respective delta aggregation task.
5. Confirm the changes have been retrieved into IdentityIQ.

Note: For SunOne-Direct and Tivoli-Direct applications, the delta aggregation task would fail even though the full aggregation is successful in case if the server has not been configured for changelog. Hence, before performing full aggregation ensure the changelog has been configured for the directory server.

**Delta aggregation for Microsoft Active Directory, ADAM,
SunOne and Tivoli**